ENKRIPSI DAN DESKRIPSI METODE DES (DATA ENCRYPTION STANDART) DENGAN MENGGUNAKAN BAHASA PEMROGRAMAN PHP

Andri Susanti (1412120006)
M. Agus Suprayitno (1412120002)
Santoso (1412120211)

ABSTRAK

Manusia hidup saling berkomunikasi antara satu dengan yang lain, di antara informasi-informasi tersebut, pastilah ada informasi yang bersifat rahasia, artinya hanya boleh diketahui oleh orang-orang tertentu saja.. Contoh lainnya banyak sekali ditemukan, seperti rahasia dagang antar perusahaan, strategi perang, hingga rahasia negara yang hanya boleh diketahui oleh petinggi suatu negara. Akan sangat berbahaya apabila rahasia-rahasia tersebut diketahui oleh pihak lain, apalagi pihak tersebut adalah musuh yang berusaha untuk mencari kekurangan dan kelemahan suatu pihak.

Dalam perkembangannya, ada berbagai metode yang digunakan dalam kriptanalisis, sesuai dengan algoritma kriptografi yang ingin dipecahkannya. Teori matematika yang digunakan dalam kriptanalisis lebih banyak daripada kriptografi, ditambahkan dengan teori probabilitas, statistika, geometri, bahasa (sastra), bahkan ada yang sampai menggunakan matematika analisis yang mempelajari konsep limit, turunan dan integral.

Maka dari itu berbagai metode *encrypt* data semakin berkembang, dari kata sandi Caesar sampai sekarang yang banyak variasi metode enkripsi-deskripsi. Salah satu metode enkrips-dekripsi data adalah DES (*Data Encryption Standard*). Yaitu proses penyandian data dengan menggunakan metode aritmatika kode binner.

Dengan adanya jurnal enkripsi-dekripsi DES ini diharapkan dapat menambah pengetahuan dengan berbagai metode peyandian data. Dan diharapkan dapat mengantisipasi celah-celah keamanan kata sandi dari para *Hacker*.

Kata kunci : DES, Enkripsi, Kata Sandi, Unirow

ABSTRACT

People always communicate with each other, such an information an the other, there must be information that is confidential, meaning that should only be known by certain people only. example is information about the exam was also included secret if before the test took place, it should only be known by the lecturer and course exam committee. Look one of example, such as inter-company trade secrets, war strategy, to state secrets that should only be known by the top brass of a country. It would be very dangerous if these secrets are known by other parties, especially the party is an enemy that is trying to find flaws and weaknesses of a party.

In the process, there are a variety of methods used in cryptanalysis, in accordance with a cryptographic algorithm that want solved. Mathematical theories used in cryptanalysis more than cryptography, is added to the theory of probability, statistics, geometry, language (literature), and some even use mathematical analysis to studied the concept of limits, derivatives and integrals.

So that various methods of growing encrypt the data, from the cipher said until now that many different methods of encryption and description. One method enkrips decrypt the data DES (Data Encryption Standard). It is the process of encoding data using arithmetic Binner code.

With journal DES encryption and decryption is expected to increase the knowledge of various methods of data peyandian. And is expected to anticipate Hacker password security gaps.

Keyword : DES, Encrypt, Password, Unirow

PENDAHULUAN

Kriptografi (atau kriptologi; dari Bahasa Yunani Kriptos, "tersembunyi, rahasia") merupakan keahlian dan ilmu dari cara-cara untuk komunikasi dengan aman pada kehadirannya dipihak ketiga. Kriptografi sebelum pada termodernisasi merupakan sinonim dari enkripsi, konversi dari kalimat-kalimat yang dapat dibaca menjadi kelihatan tidak masuk akal.

Setelah dienkrip dengan suatu kunci dinamakan ciphertext. Keamanan suatu informasi agar tidak jatuh ke tangan orangorang yang tidak berkepentingan sangatlah penting agar tidak disalahgunakan. Informasi ini dapat berupa password, nomor kartu kredit, ataupun informasi pribadi lainnya.

Bentuk awal dari penulisan rahasia membutuhkan lebih sedikit implementasi penulisan sejak banyak orang tidak dapat membaca. Penggunaan awal kriptografi yang diketahui merupak teks sandi yang diukir pada batu di Mesir (1900 sebelum Masehi).

Berbagai algoritma kriptografi telah diciptakan oleh para ahli kriptografi, namun berbagai usaha dilakukan oleh cracker untuk memecahkannya tidak sedikit yang membawa keberhasilan. Hal ini mendorong para kriptografi untuk menciptakan algoritma-algoritma yang lebih aman.

DATA ENCRYPTION STANDART

DES adalah tipikal blok chiper suatu algoritma yang membutuhkan tetap serangkaian panang dan mengubah bit plaint text melalui serangkaian operasi rumit ke bitstring cipherteks lain yang sama panjang.

Seiring dengan perkembangan teknologi, kunci DES yang sebesar 56-bit dianggap sudah tidak memadai lagi. Pada tahun 1998, 70 ribu komputer di Internet berhasil membobol satu kunci DES dalam waktu 96 hari. Tahun 1999 kejadian yang sama terjadi lagi dalam waktu lebih cepat yaitu hanya dalam waktu 22 hari. Pada tanggal 16 Juni 1998, sebuah mesin seharga 250 ribu dolar dapat dengan mudah memecahkan 25% kunci DES dalam waktu kira-kira 2,3 hari atau diperkirakan dapat memecahkan kunci DES dalam waktu 4,5 hari.

Adanya kenyataan bahwa algoritma kriptografi DES tidak lagi aman, maka NIST mulai memikirkan sebuah algoritma kriptografi lain sebagai pengganti DES. Untuk itu diadakan kontes Internasional dimana pesertanya adalah ahli kriptografi dari seluruh dunia.Adapun diadakan secara terbuka dimaksudkan agar algoritma yang baru bukan dari produk badan pemerintah yang dapat dengan sengaja menanamkan backdoor pada algoritmanya. Backdoor ini dicurigai membuat plaintext dapat langsung dibaca tanpa harus menggunakan kunci.

Pada tahun 1997 kontes pemilihan suatu standar algoritma kriptografi baru pengganti DES dimulai dan diikuti oleh 21 peserta dari seluruh dunia. Algoritma yang akan dipilih selain harus memenuhi beberapa kriteria, yaitu

a) Faktor keamanan, yang berarti algoritma tersebut harus tidak mudah dipecahkan oleh cracker, bersifat acak atau tidak mudah diterka outputnya, dan tidak berdasar algoritma matematika tertentu.
b) Faktor biaya, dimana diperhitungkan kecepatan prosesing pada baik pada hardware dan software, dan besarnya memory yang dipakai.
c) Faktor karakteristik implementasi, yakni meliputi kesederhanaan algoritma yang digunakan, kemudahan dan keamanan dalam implementasi di hardware dan software.

DES seharusnya terdiri dari algoritma enkripsi data yang diimplementasikan dalam peralatan elektronik untuk tujuan tertentu. Peralatan ini dirancang menurut cara yang

mereka gunakan dalam sistem atau jaringan komputer untuk melengkapi perlindungan cryptographic pada data biner.

Metode implementasi akan tergantung pada aplikasi dan lingkungan di sekitar sistem itu. Peralatan itu diimplementasikan tetapi sebelumnya diuji dan divalidkan secara akurat untuk menampilkan transformasi dalam bentuk algoritma.

Pada bahasan kali ini, algoritma DES akan digunakan untuk mengenkripsi data dan diimplementasikan dengan menggunakan bahasa pemrograman PHP.

DESKRIPSI ALGORITMA DES

Secara umum, algoritma DES terbagi menjadi 3 kelompok di mana kelompok yang satu dengan yang lain saling berinteraksi dan terkait antara satu dengan yang lain.

Kelompok-kelompok tersebut adalah : Pemrosesan kunci, enkripsi data 64 bit, dan dekripsi data 64 bitAlgoritma DES dirancang untuk menulis dan membaca berita blok data yang terdiri dari 64 bit di bawah kontrol kunci 64 bit. Dalam pembacaan berita harus dikerjakan dengan menggunakan kunci yang sama dengan waktu menulis berita, dengan penjadualan alamat kunci bit yang diubah sehingga proses membaca adalah kebalikan dari proses menulis.

Sebuah blok ditulis dan ditujukan pada permutasi dengan inisial IP, kemudian melewati perhitungan dan perhitungan tersebut sangat tergantung pada kunci kompleks dan pada akhirnya melewati permutasi yang invers dari permutasi dengan inisial IP-1.

Perhitungan yang tergantung pada kunci tersebut dapat didefinisikan sebagai fungsi f, yang disebut fungsi cipher dan fungsi KS, yang disebut Key Schedule.

Sebuah dekripsi perhitungan diberikan pada awal, sepanjang algoritma yang digunakan dalam penulisan pesan. Berikutnya, penggunaan algoritma untuk pembacaan pesan didekripsikan. Akhirnya, definisi dari fungsi cipher f menjadi fungsi seleksi Si dan fungsi permutasi adalah P.

Perhatikan contoh berikut ini :

Diberikan 2 blok yaitu L dan R dari bit. LR merupakan blok yang terdiri dari bit L dan diikuti oleh bit R. Kemudian menyusul urutan bit yang saling berhubungan yaitu : B1, B2, …., B8. Artinya bit yang terdiri dari B1 diikuti B2 dan akhirnya diikuti B8.

Di dalam algoritma DES dijabarkan menjadi 2 bagian, yaitu penulisan pesan dan penguraian pesan. 64 bit dari blok input yang dienkripsi adalah subjek pertama dari permutasi yang disebut permutasi dengan inisial IP. Perhatikan tabel permutasi inisial IP.

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

Input yang mengalami permutasi mempunyai bit 58 dari input bit pertamanya, bit 50 sebagai bit kedua dan bit ke 7 sebagai bit terakhir. Blok input yang mengalami permutasi kemudian menjadi input pada perhitungan dan tergantung pada kunci kompleks.

Output perhitungan ini, disebut preoutput dan output ini akan diteruskan pada permutasi berikutnya yang merupakan kebalikan dari permutasi inisial. Perhatikan tabel kebalikan dari permutasi inisial IP yaitu IP –1 .

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

Output dari algoritma di atas mempunyai bit 40 dari blok preoutput sebagai bit pertamanya, bit 8 sebagai bit kedua sampai bit 25 sebagai bit terakhir.

Perhitungan yang menggunakan blok input dikenakan permutasi sebagai inputnya untuk menghasilkan blok preoutput. Tetapi untuk pertukaran blok akhir, dari 16 iterasi dari kalkulasi yang dijelaskan di bawah ini merupakan fungsi cipher f yang mengoperasikan 2 blok, yaitu salah satu dari 32 bit dan salah satu dari 48 bit. Kalkulasi tersebut akan menghasilkan blok sepanjang 32 bit.

64 bit dari blok input terdiri dari 32 bit blok L dan diikuti oleh 32 bit blok R. Input blok ini didefinisikan sebagai LR.

K menjadi input blok dari 48 bit yang dipilih dari 64 bit kunci. Kemudian output L'R' dari iterasi dengan input LR menghasilkan persamaan berikut ini : (1)

L' = R

R' = L(+)f(R,K)

Dimana (+) merupakan penambahan bit demi bit kemudian dibagi 2.

Input iterasi pertama dari perhitungan tadi adalah blok input yang mengalami permutasi. L'R' adalah output dari iterasi ke 16, kemudian R'L'

adalah blok preoutput. Pada masing-masing iterasi sebuah blok yang berbeda, K merupakan kunci bit yang dipilih dari 64 kunci yang ditunjukkan oleh KEY.

Dengan notasi di atas, kita bisa menjelaskan iterasi menjadi lebih rinci. KS menjadi sebuah fungsi yang menggunakan bilangan bulat n dengan jangkauan dari bilangan 1 sampai bilangan 16 dan blok 64 bit KEY sebagai input serta hasilnya sebagai output blok 48 bit Kn, di mana bisa dilihat pada persamaan berikut ini : (2)

Kn = KS (n, KEY)

Dengan Kn ditentukan oleh bit dalam posisi bit yang berbeda dengan KEY. KS disebut kunci schedule karena blok K digunakan dalam iterasi ke-n (persamaan 1) dan blok Kn ditentukan oleh persamaan 2.

Karena sebelumnya blok input dipermutasikan dengan LR, akhirnya L0 dan R0 berubah menjadi L dan R, sedangkan Ln dan Rn berubah menjadi L' dan R' (persamaan 1). Selanjutnya L dan R berubah menjadi Ln-1 dan Rn-1. K adalah Kn, yaitu ketika n dalam jangkauan bilangan 1 sampai bilangan 16. Perhatikan persamaan berikut ini : (3)

Ln = Rn-1 Rn n = Ln-1 (+) f(Rn-1, Kn)

Blok preoutput dari persamaan di atas adalah R16L16.

Untuk algoritma secara rinci dapat dilihat pada bahasan enkripsi 64 bit yang akan dibahas pada bagian bawah bab ini.

Penguraian Pesan

Permutasi IP-1 menerapkan blok preoutput yang merupakan kebalikan dari permutasi dengan inisial IP. Adapun persamaan berikut ini (4) merupakan kelanjutan dari persamaan 1.

R' = L

L' = R(+) f(L', K)

Akibatnya, penguraian pesan ini harus menerapkan algoritma yang sama pada waktu pesan ditulis. Dengan mengambil masing-masing iterasi dari perhitungan blok yang sama dari kunci bit K maka penguraian itu dilakukan. Dengan menggunakan notasi-notasi dari persamaan berikut ini menjelaskan kondisi berikut : (5)

Rn-1 = Ln

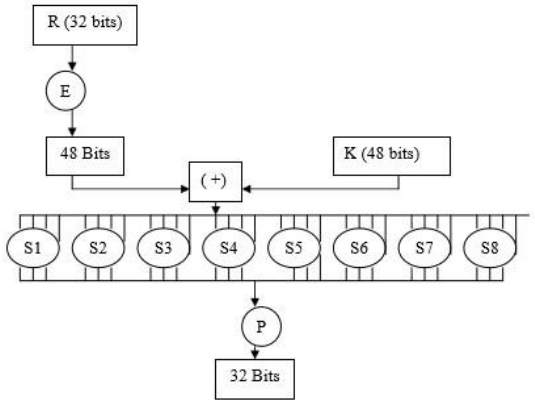
Ln-1 = Rn (+) f(Ln, Kn)

Setelah adanya persamaan di atas, sekarang R16L16 adalah blok input dari permutasi dalam perhitungan penguraian dan L0 dan R0 adalah blok preoutput. Untuk penguraian perhitungan dengan R16L16 sebagai input permutasi. K16 digunakan dalam iterasi yang pertama, K15 sebagai yang kedua dan seterusnya sampai dengan K1 digunakan dalam iterasi ke-16.

Untuk algoritma secara rinci dapat dilihat pada bahasan dekripsi 64 bit yang akan dibahas pada bagian bawah bab ini.

FUNGSI CIPHER F

Perhitungan dari fungsi f(R,K) dapat dilihat pada gambar berikut ini.



E merupakan fungsi yang mengambil blok 32 bit sebagai input dan hasil blok 48 bit sebagai output. E yang 48 bit sebagai output ditulis sebagai 8 blok dari 6 bit yang masing-masing diperoleh dengan cara menyeleksi bit dalam input. Perhatikan gambar tabel berikut ini.

TABEL E-BIT SELEKSI

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	33

3 bit pertama dari E adalah bit dalam posisi 32, kemudian 1 disusul 2 dari R dan 2 bit E yang terakhir adalah bit dalam posisi 32 dan 1.

Masing-masing fungsi seleksi untuk S1, S2, …. S8 mengambil blok 6 bit sebagai input dan hasil blok 4 bit sebagai output dan diilustrasikan dengan menggunakan tabel yang berisi S1.

S_1																
	Column Number															
Row Number	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
2	4	1	14	8	13	6	2	11	15	12	9	7	7	10	5	0
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

Jika S1 adalah fungsi yang didefinisikan dalam tabel dan B adalah blok dari 6 bit, kemudian S1 (B) ditentukan sebagai berikut : bit pertama dan terakhir dari B mewakili dalam base-2 sebuah angka dalam jangkauan 0 sampai dengan 3. Angka tersebut didefinisikan sebagai i. 4 bit ditengah dari B mewakili dalam base 2 sebuah angka dalam jangkauan 0 sampai dengan 15. Angka tersebut didefinisikan sebagai j. Lihat tabel di atas, angka dalam baris ke-i dan kolom ke-j. Angka dalam jangkauan 0 sampai dengan 15 dan diwakili oleh 4 bit blok. Blok itu adalah output S1 (B) dari S1 untuk input B.

Sebagai contoh, untuk input 011011 baris 01, baris 1 dan kolom ditentukan oleh 1101, kolom 13. Pada barus 1 kolom 13 kelihatan 5 sehingga outputnya adalah 0101.

Hasil fungsi permutasi P output 32 bit dari input 32 bit dengan permutasi bit dari input blok dapat dilihat pada tabel berikut ini :

P			
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

Output P(L) untuk fungsi P didefinisikan oleh tabel di atas dan diperoleh dari input L dengan mengambil bit ke-16 dari L sebagai bit pertama P(L), bit ke-7 sebagai bit ke-2 dari P(L), dan seterusnya sampai bit ke-25 dari L diambil sebagai bit ke-32 dari P(L).

S1, S2, ….., S8 menjadi fungsi seleksi yang berbeda dari P menjadi fungsi permutasi sekaligus E menjadi fungsi yang telah didefinisikan di atas.

Untuk mendefinisikan f(R,K), langkah pertama adalah mendefinisikan B1,…B8 menjadi blok dari 6 bit masing-masing untuk persamaan di bawah ini : (6) B1, B2, …, B8 = K(+) E (R)

Blok f(R,K) kemudian didefinisikan menjadi persamaan berikut ini : (7) P(S1(B1)(S2(B2)…. (S8(B8))

Jadi K (+) E (R) adlaha hasil pertama yang dibagi dalam 8 blok input yang dapat dilihat pada persamaan (6). Kemudian masing-masing B1 diambil sebagai input untuk S1 dan 8 blok (S1(B1)(S2(B2)…. (S8(B8)) dari 4 bit masing-masing digabungkan menjadi blok tunggal dari 32 bit yang membentuk input P. Output pada persamaan (7) kemudian menjadi input bagi R dan K.

PEMROSESAN KUNCI

Sebelum kita membuat diagram blok tentang alur pemrosesan kunci, sebelumnya disusun terlebih dahulu algoritma yang menunjang adanya pemrosesan kunci. Algoritma ini nantinya akan sangat berguna sekali pada waktu implementasi pada program.

Adapun algoritmanya adalah sebagai berikut : - Pengguna memasukkan sebuah kunci sebesar 64 bit atau 8 karakter, dimana nantinya setiap bit dalam kunci ini akan digunakan bit paritas - Sebelum dilakukan permutasi terhadap kunci tersebut, perlu diadakan penjadwalan kunci rahasia (secret key-schedulling). Hal ini dilakukan untuk menyusun 16 buah kunci yang akan dimasukkan pada setiap iterasi DES, baik pada enkripsi maupun dekripsi. - Setelah langkah ke-2 selesai, dilanjutkan dengan permutasi. Permutasi dilakukan pada kunci 64 bit tadi. Pada tahapan ini, bit-bit paritas tidak dilibatkan sehingga bit kunci berkurang menjadi 56 bit. Bit 1 pada kunci ke-56 merupakan bit 57 pada kunci awalnya, bit 2 adalah bit 49, dan seterusnya hingga bit 56 adalah bit 4 kunci 64. Posisi bit hasil permutasi pada langkah pertama ini diberi nama Permuted Choice 1 (PC-1). Adapun hasilnya dapat dilihat pada keterangan di bawah ini :

Permuted Choice 1 (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	46	38	30	22
14	6	61	53	45	37	29
21	13	5	28	20	12	4

- Dari permutasi ini kemudian output PC-1 dibagi menjadi 2 bagian yaitu 28 bit pertama disebut C(0) dan 28 bit terakhir disebut D(0). - Dari C(0) dan D(0) kemudian dihitung sub-sub kunci untuk setiap iterasi , yang dimulai dengan j = 1.

- Untuk setiap iterasi, yaitu j rotasi ke kiri 1 kali atau sebanyak 2 kali untuk setiap C(j – 1) dan D(j – 1). Dari hasil rotasi ini akan didapatkan hasil C(j) dan D(j). Tabel berikut ini akan menunjukkan langkah setiap rotasi yang diterapkan pada setiap iterasinya.

Iterasi ke

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

Jumlah Step

1 1 2 2 2 2 2 2 1 2 2 2 2 2 2 1

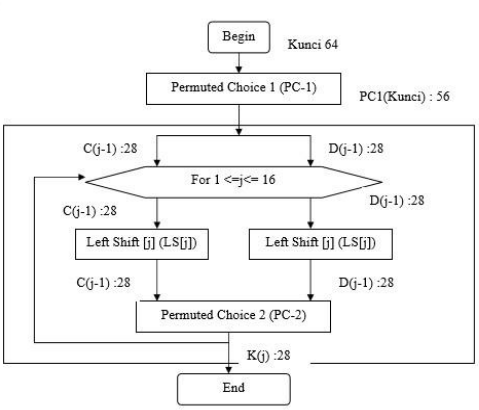
- Untuk setiap hasil C(j) dan D(j), kunci pada iterasi ke j didapatkan dengan cara melakukan permutasi kembali pada C(j) dan D(j). Permutasi itu dikenal dengan nama Permuted Choice (PC-2). Perhatikan hasilnya berikut ini :

Permuted Choice 1 (PC-1)

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

- Iterasi dilakukan terus menerus hingga ke-16 kunci berhasil disusun.

Adapun diagram blok dari pemrosesan kunci yang sudah dibuat algoritma di atas, dapat dilihat pada gambar di bawah ini.



Contoh Enkripsi Data Dengan Algoritma DES

Langkah-langkah mengenkripsi data menggunakan algoritma DES(Data Encryption System) yaitu:

Diberikan contoh:

- Plaintext(x) = COMPUTER
- Key(k) = 13 34 57 79 9B BC DF F1

Langkah Pertama :

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	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

Urutan bit pada plaintext urutan ke 58 ditaruh diposisi 1,

Urutan bit pada plaintext urutan ke 50 ditaruh di posisi 2,

Urutan bit pada plaintext urutan ke 42 ditaruh di posisi 3, dst

Sehingga hasil outputnya adalah

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

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

L^0 : 11111111 10111000 01110110 01010111 (tabel IP dengan warna kuning)

R^0 : 00000000 00000000 00000110 10000011 (tabel IP dengan warna hijau)

Langkah Ketiga :

Generate kunci yang akan digunakanuntuk 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 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 : 1111000 0110011 0010101 0101111

(tabel PC-1 warna kuning)
D_0 : 0101010 1011001 1001111 0001111

(tabel PC-1 warna hijau)

Langkah Keempat :

Lakukan pergeseran kiri (Left Shift) 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 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

Untuk putaran ke 1, dilakukan pegeseran 1 bit ke kiri

Untuk putaran ke 2, dilakukan pergeseran 1 bit kekiri

Untuk putaran ke 3, dilakukan pergeseran 2 bit kekiri, dst

Berikut hasil outputnya:

C_0 : 1111000 0110011 0010101 0101111

D_0 : 0101010 1011001 1001111 0001111

Digeser 1 bit ke kiri

C_1 : 1110000 1100110 0101010 1011111

D_1 : 1010101 0110011 0011110 0011110

Digeser 2 bit ke kiri

C_2 : 1100001 1001100 1010101 0111111

D_2 : 0101010 1100110 0111100 0111101

Digeser 2 bit ke kiri

C_3 : 0000110 0110010 1010101 1111111

D_3 : 0101011 0011001 1110001 1110101

Digeser 2 bit ke kiri

C_4 : 0011001 1001010 1010111 1111100

D_4 : 0101100 1100111 1000111 1010101

Digeser 2 bit ke kiri

C_5 : 1100110 0101010 1011111 1110000

D_5 : 0110011 0011110 0011110 1010101

Digeser 2 bit ke kiri

C_6 : 0011001 0101010 1111111 1000011

D_6 : 1001100 1111000 1111010 1010101

Digeser 2 bit ke kiri

C_7 : 1100101 0101011 1111110 0001100

D_7 : 0110011 1100011 1101010 1010110

Digeser 2 bit ke kiri

C_8 : 0010101 0101111 1111000 0110011

D_8 : 1001111 0001111 0101010 1011001

Digeser 1 bit ke kiri

C_9 : 0101010 1011111 1110000 1100110

D_9 : 0011110 0011110 1010101 0110011

Digeser 2 bit ke kiri

C_{10} : 0101010 1111111 1000011 0011001

D_{10} : 1111000 1111010 1010101 1001100

Digeser 2 bit ke kiri

C_{11} : 0101011 1111110 0001100 1100101

D_{11} : 1100011 1101010 1010110 0110011

Digeser 2 bit ke kiri

C_{12} : 0101111 1111000 0110011 0010101

D_{12} : 0001111 0101010 1011001 1001111

Digeser 2 bit ke kiri

C_{13} : 0111111 1100001 1001100 1010101

D_{13} : 0111101 0101010 1100110 0111100

Digeser 2 bit ke kiri

C_{14} : 1111111 0000110 0110010 1010101

D_{14} : 1110101 0101011 0011001 1110001

Digeser 2 bit ke kiri

C_{15} : 1111100 0011001 1001010 1010111

D_{15} : 1010101 0101100 1100111 1000111

Digeser 1 bit ke kiri

C_{16} : 1111000 0110011 0010101 0101111

D_{16} : 0101010 1011001 1001111 0001111

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 CiDi 48 bit.

Tabel 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:

C_1D_1 = 1110000 1100110 0101010 1011111 1010101 0110011 0011110 0011110

K_1 = 000110 110000 001011 101111 111111 000111 000001 110010

C_2D_2 = 1100001 1001100 1010101 0111111 0101010 1100110 0111100 0111101

K_2 = 011110 011010 111011 011001 110110 111100 100111 100101

C_3D_3 = 0000110 0110010 1010101 1111111 0101011 0011001 1110001 1110101

K_3 = 010101 011111 110010 001010 010000 101100 111110 011001

C_4D_4 = 0011001 1001010 1010111 1111100 0101100 1100111 1000111 1010101

K_4 = 011100 101010 110111 010110 110110 110011 010100 011101

C_5D_5 = 1100110 0101010 1011111 1110000 0110011 0011110 0011110 1010101

K_5 = 011111 001110 110000 000111 111010 110101 001110 101000

C_6D_6 = 0011001 0101010 1111111 1000011 1001100 1111000 1111010 1010101

K_6 = 011000 111010 010100 111110 010100 000111 101100 101111

C_7D_7 = 1100101 0101011 1111110 0001100 0110011 1100011 1101010 1010110

K_7 = 111011 001000 010010 110111 111101 100001 100010 111100

C_8D_8 = 0010101 0101111 1111000 0110011 1001111 0001111 0101010 1011001

K_8 = 111101 111000 101000 111010 110000 010011 101111 111011

C_9D_9 = 0101010 1011111 1110000 1100110 0011110 0011110 1010101 0110011

K_9 = 111000 001101 101111 101011 111011 011110 011110 000001

$C_{10}D_{10}$ = 0101010 1111111 1000011 0011001 1111000 1111010 1010101 1001100

K_{10} = 101100 011111 001101 000111 101110 100100 011001 001111

$C_{11}D_{11}$ = 0101011 1111110 0001100 1100101 1100011 1101010 1010110 0110011

K_{11} = 001000 010101 111111 010011 110111 101101 001110 000110

$C_{12}D_{12}$ = 0101111 1111000 0110011 0010101 0001111 0101010 1011001 1001111

K_{12} = 011101 010111 000111 110101 100101 000110 011111 101001

$C_{13}D_{13}$ = 0111111 1100001 1001100 1010101 0111101 0101010 1100110 0111100

K_{13} = 100101 111100 010111 010001 111110 101011 101001 000001

$C_{14}D_{14}$ = 1111111 0000110 0110010 1010101 1110101 0101011 0011001 1110001

K_{14} = 010111 110100 001110 110111 111100 101110 011100 111010

$C_{15}D_{15}$ = 1111100 0011001 1001010 1010111 1010101 0101100 1100111 1000111

K_{15} = 101111 111001 000110 001101 001111 010011 111100 001010

$C_{16}D_{16}$ = 1111000 0110011 0010101 0101111 0101010 1011001 1001111 0001111

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

Langkah Kelima :

Pada langkah ini, 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 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 Ki dan menghasilkan Vektor Matriks Ai.

Berikut hasil outputnya:

Iterasi 1

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

K_1 = 000110 110000 001011 101111 111111 000111 000001 110010

---------- XOR

A_1 = 100110 110000 001011 101111 111111 001010 010001 110100

Berhubung bagian dibawah ini yang paling ribet, maka saya tambahkan keterangan ditengah-tengah proses iterasi. Bisa kita lihat pada iterasi1 diatas setelah kita dapatkan 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 ke dalam 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) = 011010 101110 100001 010110 100110 100101 010000 001101

K2 = 011110 011010 111011 011001 110110 111100 100111 100101

---------- XOR

A2 = 000100 110100 011010 001111 010000 011001 110111 101000

Iterasi – 3

E(R(3)-1) = 010001 010111 111011 110011 110001 010101 010010 100001

K3 = 010101 011111 110010 001010 010000 101100 111110 011001

---------- XOR

A3 = 000100 001000 001001 111001 100001 111001 101100 111000

Iterasi – 4

E(R(4)-1) = 010111 110001 010111 110011 110101 011100 001111 110001

K4 = 011100 101010 110111 010110 110110 110011 010100 011101

---------- XOR

A4 = 001011 011011 100000 100101 000011 101111 011011 101100

Iterasi – 5

E(R(5)-1) = 110110 101001 011100 000101 011001 011010 100110 100011

K5 = 011111 001110 110000 000111 111010 110101 001110 101000

---------- XOR

A5 = 101001 100111 101100 000010 100011 101111 101000 001011

Iterasi – 6

E(R(6)-1) = 100101 011011 110001 010110 101110 101100 000111 111010

K6 = 011000 111010 010100 111110 010100 000111 101100 101111

---------- XOR

A6 = 111101 100001 100101 101000 111010 101011 101011 010101

Iterasi – 7

E(R(7)-1) = 110010 100001 011111 110010 100111 111101 011001 010011

K7 = 111011 001000 010010 110111 111101 100001 100010 111100

---------- XOR

A7 = 001001 101001 001101 000101 011010 011100 111011 101111

Iterasi – 8

E(R(8)-1) = 111100 001010 101001 010101 010011 110000 001010 100011

K8 = 111101 111000 101000 111010 110000 010011 101111 111011

---------- XOR

A8 = 000001 110010 000001 101111 100011 100011 100101 011000

Iterasi – 9

E(R(9)-1) = 010010 101111 111000 000000 000010 101111 110101 010001

K9 = 111000 001101 101111 101011 111011 011110 011110 000001

---------- XOR

A9 = 101010 100010 010111 101011 111001 110001 101011 010000

Iterasi – 10

E(R(10)-1)= 100111 111000 001110 100010 100111 110111 111000 001010

K10 = 101100 011111 001101 000111 101110 100100 011001 001111

---------- XOR

A10 = 001011 100111 000011 100101 001001 010011 100001 000101

Iterasi – 11

E(R(11)-1)= 010011 110111 111010 101010 101111 110011 110001 011001

K11 = 001000 010101 111111 010011 110111 101101 001110 000110

---------- XOR

A11 = 011011 100010 000101 111001 011000 011110 111111 011111

Iterasi – 12

E(R(12)-1)= 001001 011010 101001 011111 110001 010111 110010 101100

K12 = 011101 010111 000111 110101 100101 000110 011111 101001

---------- XOR

A12 = 010100 001101 101110 101010 010100 010001 101101 000101

Iterasi – 13

E(R(13)-1)= 100110 100111 110111 111011 111110 101110 101100 001010

K13 = 100101 111100 010111 010001 111110 101011 101001 000001

---------- XOR

A13 = 000011 011011 100000 101010 000000 000101 000101 001011

Iterasi – 14

E(R(14)-1)= 111001 010111 110000 001000 001000 001000 001011 111011

K14 = 010111 110100 001110 110111 111100 101110 011100 111010

---------- XOR

A14 = 101110 100011 111110 111111 110100 100110 010111 000001

Iterasi – 15

E(R(15)-1)= 000110 101100 001100 000001 011001 011010 100101 010100

K15 = 101111 111001 000110 001101 001111 010011 111100 001010

---------- XOR

A15 = 101001 010101 001010 001100 010110 001001 011001 011110

Iterasi – 16

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

K16 = 110010 110011 110110 001011 000011 100001 011111 110101

---------- XOR

A16 = 011111 101110 100010 001110 010110 110000 001001 010111

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 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

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
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

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:

- Bit pertama dan terakhir yaitu 1 dan 0 digabungkan menjadi 10
- Bit kedua hingga ke lima 0011

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:

B_1 = 1000 0101 0100 1000 0011 0010 1110 1010

B_2 = 1101 1100 0100 0011 1000 0000 1111 1001

B_3 = 1101 0110 0011 1100 1011 0110 0111 1111

B_4 = 0010 1001 1101 0000 1011 1010 1111 1110

B^5 = 0100 0001 0011 1101 1000 1010 1100 0011

B_6 = 0110 1101 1101 1100 0011 0101 0100 0110

B_7 = 1110 0011 0110 1011 0000 0101 0010 1101

B_8 = 0000 1000 1101 1000 1000 0011 1101 0101

B_9 = 0110 1110 1110 0001 1010 1011 0100 1010

B_{10} = 0010 0001 0111 0000 0100 0001 0110 1101

B_{11} = 0101 1110 0000 1100 1101 1011 1100 0010

B_{12} = 0110 1000 0000 1011 0011 0110 1010 1101

B_{13} = 1111 1001 1101 1011 0010 0100 1011 0011

B_{14} = 1011 1000 0111 1110 1100 0101 1100 0001

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

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

Langkah Ketujuh:

Setelah didapatkan nilai vektor Bi, langkah selanjutnya adalah memutasikan bit vektor 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:

$P(B_1)$ = 00101000 10110011 01000100 11010001

$P(B_2)$ = 10001011 11011001 10001100 00010011

$P(B_3)$ = 01101111 10110010 10011100 11111110

$P(B_4)$ = 00111111 00111011 01000111 10100001

$P(B_5)$ = 10010101 00110010 11011000 01000101

$P(B_6)$ = 00100100 00011011 11110011 11111000
$P(B_7)$ = 11001000 11000001 11101110 01101100

$P(B_8)$ = 00000111 00111001 00101001 01100001

$P(B_9)$ = 11011001 00111011 10100011 10010100

$P(B_{10})$ = 00001100 00010101 01101110 00100100

$P(B_{11})$ = 01110001 00111110 10110000 01010011

$P(B_{12})$ = 10101000 01101000 10001110 11101001

$P(B_{13})$ = 10000110 11001011 11001111 11001011

$P(B_{14})$ = 00000101 11011101 00111010 01001111

$P(B_{15})$ = 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.

L0 = 11111111 10111000 01110110 01010111

R0 = 00000000 00000000 00000110 10000011

P(B1) = 00101000 10110011 01000100 11010001

L(1)-1 = 11111111 10111000 01110110 01010111

----------XOR

R1 = 11010111 00001011 00110010 10000110

P(B2) = 10001011 11011001 10001100 00010011

L(2)-1 = 00000000 00000000 00000110 10000011

----------XOR

R2 = 10001011 11011001 10001010 10010000

P(B3) = 01101111 10110010 10011100 11111110

L(3)-1 = 11010111 00001011 00110010 10000110

----------XOR

R3 = 10111000 10111001 10101110 01111000

P(B4) = 00111111 00111011 01000111 10100001

L(4)-1 = 10001011 11011001 10001010 10010000

----------XOR

R4 = 10110100 11100010 11001101 00110001

P(B5) = 10010101 00110010 11011000 01000101

L(5)-1 = 10111000 10111001 10101110 01111000

----------XOR

R5 = 00101101 10001011 01110110 00111101

P(B6) = 00100100 00011011 11110011 11111000

L(6)-1 = 10110100 11100010 11001101 00110001

----------XOR

R6 = 10010000 11111001 00111110 11001001

P(B7) = 11001000 11000001 11101110 01101100

L(7)-1 = 00101101 10001011 01110110 00111101

----------XOR

R7 = 11100101 01001010 10011000 01010001

P(B8) = 00000111 00111001 00101001 01100001

L(8)-1 = 10010000 11111001 00111110 11001001

----------XOR

R8 = 10010111 11000000 00010111 10101000

P(B9) = 11011001 00111011 10100011 10010100

L(9)-1 = 11100101 01001010 10011000 01010001

----------XOR

R9 = 00111100 01110001 00111011 11000101

P(B10) = 00001100 00010101 01101110 00100100

L(10)-1 = 10010111 11000000 00010111 10101000

----------XOR

R10 = 10011011 11010101 01111001 10001100

P(B11) = 01110001 00111110 10110000 01010011

L(11)-1 = 00111100 01110001 00111011 11000101

----------XOR

R11 = 01001101 01001111 10001011 10010110

P(B12) = 10101000 01101000 10001110 11101001

L(12)-1 = 10011011 11010101 01111001 10001100

----------XOR

R12 = 00110011 10111101 11110111 01100101

P(B13) = 10000110 11001011 11001111 11001011

L(13)-1 = 01001101 01001111 10001011 10010110

----------XOR

R13 = 11001011 10000100 01000100 01011101

P(B14) = 00000101 11011101 00111010 01001111

L(14)-1 = 00110011 10111101 11110111 01100101

----------XOR

R14 = 00110110 01100000 11001101 00101010

P(B15) = 10100101 00100110 11101100 11101100

L(15)-1 = 11001011 10000100 01000100 01011101

----------XOR

R15 = 01101110 10100010 10101000 10110001

P(B16) = 00101001 11110111 01101000 11001100

L(16)-1 = 00110110 01100000 11001101 00101010

----------XOR

R16 = 00011111 10010111 10100101 11100110

L16 = 01101110 10100010 10101000 10110001

Langkah Kedelapan:

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

Tabel 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

Sehingga Input :

$R_{16}L_{16}$ = 00011111 10010111 10100101 11100110 01101110 10100010 10101000 10110001

Menghasilkan Output:

Cipher(dalam biner) = 01010110 11110001 11010101 11001000 01010010 10101111 10000001 00111111

atau

Cipher(dalam hexa) = 56 f1 d5 c8 52 af 81 3f

Penyandian Metode DES di PHP sederhana.

Index.php

```
<html>
<head>
<title>Enkripsi DES</title>
</head>
<body>
<form id="form1" name="form1" method="post" action="enkripsi.php">
TEXT:<input type="text" name="enkripsi">
<br>KEY:<input type="text" name="key">
<input type="submit" name="submit" value="Enkripsi">
</form>
</body>
</html>
```

enkripsi.php

```
<?php
$str=$_POST['enkripsi'];
$key=$_POST['key'];
$crypt=crypt($str);
$des=crypt($str, $key);
$base64enkripsi=base64_encode($str);
$base64dekripsi=base64_decode($base64enkripsi);
echo"$crypt";
echo"<br>$des";
echo"<br>$base64enkripsi";
echo"<br>$base64dekripsi";
?>
```

Referensi

Liddell and Scott's Greek-English Lexicon. Oxford University Press. 1984

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

http://www.britannica.com/topic/Data-Encryption-Standard