

Nama : Ridayanti Wardani

NRP : 152023168

Tugas : Keamanan Jaringan (DES)

Plaintext: RIDAAAAA

Kunci : 13 34 57 79 9B BC DF F1

Lakukan enkripsi dengan DES

Langkah 1 (Mengubah Plaintext dan kunci menjadi bilangan biner)

Plaintext	Hexa	Binner							
R	52	1	2	3	4	5	6	7	8
		0	1	0	1	0	0	1	0

Plaintext	Hexa	Binner							
I	49	9	10	11	12	13	14	15	16
		0	1	0	0	1	0	0	1

Plaintext	Hexa	Binner							
D	44	17	18	19	20	21	22	23	24
		0	1	0	0	0	1	0	0

Plaintext	Hexa	Binner							
A	41	25	26	27	28	29	30	31	32
		0	1	0	0	0	0	0	1

Plaintext	Hexa	Binner							
A	41	33	34	35	36	37	38	39	40
		0	1	0	0	0	0	0	1

Plaintext	Hexa	Binner							
A	41	41	42	43	44	45	46	47	48
		0	1	0	0	0	0	0	1

Plaintext	Hexa	Binner							
A	41	49	50	51	52	53	54	55	56
		0	1	0	0	0	0	0	1

Plaintext	Hexa	Binner							
A	41	57	58	59	60	61	62	63	64
		0	1	0	0	0	0	0	1

Hexa	Binner							
13	1	2	3	4	5	6	7	8
	0	0	0	1	0	0	1	1

Hexa	Binner							
34	9	10	11	12	13	14	15	16
	0	0	1	1	0	1	0	0

Hexa	Binner							
57	17	18	19	20	21	22	23	24
	0	1	0	1	0	1	1	1

Hexa	Binner							
79	25	26	27	28	29	30	31	32
	0	1	1	1	1	0	0	1

Hexa	Binner							
9B	33	34	35	36	37	38	39	40
	1	0	0	1	1	0	1	1

Hexa	Binner							
BC	41	42	43	44	45	46	47	48
	1	0	1	1	1	1	0	0

Hexa	Binner							
DF	49	50	51	52	53	54	55	56
	1	1	0	1	1	1	1	1

Hexa	Binner							
F1	57	58	59	60	61	62	63	64
	1	1	1	1	0	0	0	1

Langkah 2
Initial Permutation (IP) pada plaintext

Plaintext (X)							
1	2	3	4	5	6	7	8
0	1	0	1	0	0	1	0

Tabel IP							
1	2	3	4	5	6	7	8
58	50	42	34	26	18	10	2

IP (X)							
1	2	3	4	5	6	7	8
1	1	1	1	1	1	1	1

9	10	11	12	13	14	15	16
0	1	0	0	1	0	0	1

9	10	11	12	13	14	15	16
60	52	44	36	28	20	12	4

9	10	11	12	13	14	15	16
0	0	0	0	0	0	0	1

17	18	19	20	21	22	23	24
0	1	0	0	0	1	0	0

17	18	19	20	21	22	23	24
62	54	46	38	30	22	14	6

17	18	19	20	21	22	23	24
0	0	0	0	0	1	0	0

25	26	27	28	29	30	31	32
0	1	0	0	0	0	0	1

25	26	27	28	29	30	31	32
64	56	48	40	32	24	16	8

25	26	27	28	29	30	31	32
1	1	1	1	1	0	1	0

33	34	35	36	37	38	39	40
0	1	0	0	0	0	0	1

33	34	35	36	37	38	39	40
57	49	41	33	25	17	9	1

33	34	35	36	37	38	39	40
0	0	0	0	0	0	0	0

41	42	43	44	45	46	47	48
0	1	0	0	0	0	0	1

41	42	43	44	45	46	47	48
59	51	43	35	27	19	11	3

41	42	43	44	45	46	47	48
0	0	0	0	0	0	0	0

49	50	51	52	53	54	55	56
0	1	0	0	0	0	0	1

49	50	51	52	53	54	55	56
61	53	45	37	29	21	13	5

49	50	51	52	53	54	55	56
0	0	0	0	0	0	1	0

57	58	59	60	61	62	63	64
0	1	0	0	0	0	0	1

57	58	59	60	61	62	63	64
63	57	47	39	31	23	15	7

57	58	59	60	61	62	63	64
0	0	0	0	0	0	0	1

Hasil : IP(X) = 11111111 00000001 00000100 11111010 00000000 00000000 00000010 00000001
 Lo : 11111111 00000001 00000100 11111010
 Ro : 00000000 00000000 00000010 00000001

Langkah 3

Generate Kunci menggunakan table permutasi kompresi PC-1
Kompresi 64 bit menjadi 56 bit dengan membuang 1 bit (parity bit) tiap blok kunci

Kunci							
1	2	3	4	5	6	7	8
0	0	0	1	0	0	1	1

Tabel PC-1						
1	2	3	4	5	6	7
57	49	41	33	25	17	9

Tabel PC-1						
1	2	3	4	5	6	7
1	1	1	1	0	0	0

9	10	11	12	13	14	15	16
0	0	1	1	0	1	0	0

8	9	10	11	12	13	14
1	58	50	42	34	26	18

8	9	10	11	12	13	14
0	1	1	0	0	1	1

17	18	19	20	21	22	23	24
0	1	0	1	0	1	1	1

15	16	17	18	19	20	21
10	2	59	51	43	35	27

15	16	17	18	19	20	21
0	0	1	0	1	0	1

25	26	27	28	29	30	31	32
0	1	1	1	1	0	0	1

22	23	24	25	26	27	28
19	11	3	60	52	44	36

22	23	24	25	26	27	28
0	1	0	1	1	1	1

33	34	35	36	37	38	39	40
1	0	0	1	1	0	1	1

29	30	31	32	33	34	35
63	55	47	39	31	23	15

29	30	31	32	33	34	35
0	1	0	1	0	1	0

41	42	43	44	45	46	47	48
1	0	1	1	1	1	0	0

36	37	38	39	40	41	42
7	62	54	46	38	30	22

36	37	38	39	40	41	42
1	0	1	1	0	0	1

49	50	51	52	53	54	55	56
1	1	0	1	1	1	1	1

43	44	45	46	47	48	49
14	6	61	53	45	37	29

43	44	45	46	47	48	49
1	0	0	1	1	1	1

1	2	3	4	5	6	7	8
1	1	1	1	0	0	0	1

50	51	52	53	54	55	56
21	13	5	28	20	12	4

50	51	52	53	54	55	56
0	0	0	1	1	1	1

Hasil : CoDo = 1111000 0110011 0010101 0101111 0101010 1011001 1001111 0001111

Co : 1111000 0110011 0010101 0101111

Do : 0101010 1011001 1001111 0001111

Langkah 4 (Left Shift Operztion)

Lakukan pergeseran pada C0 dan D0 menggunakan tabel pergeseran bit 16 putaran

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

CoDo = 11110000 0110011 0010101 0101111 0101010 1011001 1001111 0001111

Co	1111000011001100101010101111	D0	0101010101100110011110001111
C1	1110000110011001010101011111	D1	1010101011001100111100011110
C2	1100001100110010101010111111	D2	0101010110011001111000111101
C3	0000110011001010101011111111	D3	0101011001100111100011110101
C4	0011001100101010101111111100	D4	0101100110011110001111010101
C5	1100110010101010111111110000	D5	0110011001111000111101010101
C6	0011001010101011111111000011	D6	1001100111100011110101010101
C7	1100101010101111111100001100	D7	0110011110001111010101010110
C8	0010101010111111110000110011	D8	1001111000111101010101011001
C9	01010101011111111100001100110	D9	0011110001111010101010110011
C10	01010101111111110000110011001	D10	1111000111101010101011001100
C11	0101011111111000011001100101	D11	1100011110101010101100110011
C12	0101111111100001100110010101	D12	0001111010101010110011001111
C13	0111111110000110011001010101	D13	0111101010101011001100111100
C14	111111000011001100101010101	D14	1110101010101100110011110001
C15	1111100001100110010101010111	D15	1010101010110011001111000111
C16	1111000011001100101010101111	D16	0101010101100110011110001111

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 1100001 1110101

K_3 = 010101 011111 110010 001010 010000 101100 111110 011001

Langkah 4 (Left Shift Operztion)

Lakukan pergeseran pada C0 dan D0 menggunakan tabel pergeseran bit 16 putaran

$C_4D_4 = 0011001\ 1001010\ 1010111\ 1111100\ 0101100\ 1100111\ 1000111\ 1010101$

$K_4 = 011100\ 101010\ 110111\ 010110\ 110110\ 110011\ 010100\ 011101$

$C_5D_5 = 1100110\ 1010101\ 1011111\ 1110000\ 0110011\ 0011110\ 0011110\ 1010101$

$K_5 = 011111\ 001110\ 110000\ 000111\ 111010\ 110101\ 011110\ 101000$

$C_6D_6 = 0011001\ 0101010\ 1111111\ 1000011\ 1001100\ 1111000\ 1110100\ 1010101$

$K_6 = 011000\ 111010\ 010100\ 111110\ 010100\ 000111\ 101100\ 101111$

$C_7D_7 = 0100101\ 0101011\ 1111110\ 0001101\ 1100011\ 1100101\ 1010110\ 1010110$

$K_7 = 111011\ 010001\ 111011\ 111101\ 110001\ 100001\ 101100\ 111100$

$C_8D_8 = 0010101\ 0111111\ 1111000\ 0110011\ 1001111\ 0011111\ 0110101\ 1011001$

$K_8 = 111101\ 111000\ 101000\ 111010\ 110000\ 010011\ 101011\ 011011$

$C_9D_9 = 0101010\ 0111111\ 1110000\ 1100100\ 1110101\ 1010101\ 1010101\ 1010101$

$K_9 = 111000\ 001101\ 101011\ 101011\ 011011\ 011111\ 011101\ 111111$

$C_{10}D_{10} = 0101010\ 1111111\ 1000011\ 0110011\ 1110101\ 0110101\ 1010101\ 1010101$

$K_{10} = 101100\ 011111\ 001101\ 001111\ 111101\ 010101\ 010101\ 010111$

$C_{11}D_{11} = 0101011\ 1111110\ 0001101\ 1100101\ 1010101\ 1010101\ 1010101\ 1010101$

$K_{11} = 001000\ 010101\ 111111\ 001011\ 101011\ 110011\ 011110\ 000011$

$C_{12}D_{12} = 0101111\ 1110000\ 0110011\ 0010111\ 0100101\ 1010101\ 1010101\ 1011111$

$K_{12} = 011101\ 010111\ 000111\ 011101\ 100111\ 011011\ 011111\ 011111$

$C_{13}D_{13} = 0111111\ 1000001\ 1001100\ 0101010\ 0111101\ 1010101\ 1110100\ 1110100$

$K_{13} = 100101\ 111000\ 011111\ 010001\ 111011\ 101001\ 000101\ 000011$

$C_{14}D_{14} = 1111111\ 0001010\ 1010010\ 1111011\ 0110010\ 1010101\ 1011111\ 1011111$

Langkah 4 (Left Shift Operztion)

Lakukan pergeseran pada C0 dan D0 menggunakan tabel pergeseran bit 16 putaran

$K_{14} = 011011\ 111000\ 011110\ 101111\ 101010\ 101111\ 111011\ 111001$

$C_{15}D_{15} = 1111110\ 0011001\ 1010100\ 1011111\ 0110011\ 1010101\ 1010101\ 1010101$

$K_{15} = 101111\ 111001\ 001111\ 101001\ 111111\ 101010\ 101010\ 101111$

$C_{16}D_{16} = 1110000\ 0110011\ 0110011\ 1010010\ 1011111\ 1010101\ 1010101\ 1010101$

$K_{16} = 110010\ 110011\ 101011\ 101001\ 101001\ 101001\ 101001\ 101001$

Langkah 5 meng-ekspansi data dari R_{i-1} 32 bit menjadi R_i 48 bit
 Sebanyak 16 kali putaran dengan nilai putaran $1 \leq i \leq 16$ menggunakan table Ekspansi

Tabel Ekspansi

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 $E(R_{i-1})$ Kemudian di XOR dengan K_i dan menghasilkan vector matriks A_i

Berikut hasil outputnya:

Iterasi 1

$R_0 = 00000000\ 00000000\ 00000010\ 00000001$

$E(R(1)-1) = 100000\ 000000\ 000000\ 000000\ 000000\ 000100\ 000000\ 000010$

$K_1 = 000110\ 110000\ 001011\ 101111\ 111111\ 000111\ 000001\ 110010$

----- XOR

$A_1 = 100110\ 110000\ 001011\ 101111\ 111111\ 000011\ 000001\ 110000$

$R_1 = 11001101\ 10110000\ 01100100\ 00100011$

$E(R(2)-1) = 111001\ 011011\ 110110\ 100000\ 001100\ 001000\ 000100\ 000111$

$K_2 = 011110\ 011010\ 111011\ 011001\ 110110\ 111100\ 100111\ 100101$

-----XOR

$A_2 = 100111\ 000001\ 001101\ 111001\ 111010\ 110100\ 100011\ 100010$

$R_2 = 01101010\ 00000011\ 01011100\ 11001100$

$E(R(3)-1) = 001101\ 010100\ 000000\ 000110\ 101011\ 111001\ 011001\ 011000$

$K_3 = 010101\ 011111\ 110010\ 001010\ 010000\ 101100\ 111110\ 011001$

-----XOR

$A_3 = 011000\ 001011\ 110010\ 001100\ 111011\ 010101\ 100111\ 000001$

$R_3 = 00011001\ 10110100\ 11001100\ 01101000$

$E(R(4)-1) = 000011\ 110011\ 110110\ 101001\ 011001\ 011000\ 001101\ 010000$

$K_4 = 011100\ 101010\ 110111\ 010110\ 110110\ 110011\ 010100\ 011101$

-----XOR

Langkah 5 meng-ekspansi data dari R_{i-1} 32 bit menjadi R_i 48 bit
Sebanyak 16 kali putaran dengan nilai putaran $1 \leq i \leq 16$ menggunakan table Ekspansi

A4 = 011111 011001 000001 111111 101111 101011 011001 001101

R4 = 00001111 11000100 01100001 10110100

E(R(5)-1) = 000001 011111 111000 001000 001100 000011 110110 101000

K5 = 011111 001110 110000 000111 111010 110101 011110 101000

-----XOR

A5 = 011110 010001 001000 001111 110110 110110 111000 000000

R5 = 10100001 11011001 01000110 01011110

E(R(6)-1) = 010100 000011 111011 110010 101000 001100 001011 111101

K6 = 011000 111010 010100 111110 010100 000111 101100 101111

-----XOR

A6 = 001100 111001 101111 001100 111100 001011 100111 010010

R6 = 10010011 01000101 01101011 11111011

E(R(7)-1) = 110010 100110 101000 001010 101101 010111 111111 110111

K₇ = 111011 010001 111011 111101 110001 100001 101100 111100

-----XOR

A7 = 001001 101110 111010 111101 010000 110110 011101 001011

R7 = 10110000 00111011 10001101 01011011

E(R(8)-1) = 110110 100000 000111 110111 110001 011010 101011 110111

K₈ = 111101 111000 101000 111010 110000 010011 101011 011011

-----XOR

A8 = 001011 011000 101111 001101 000001 001001 000100 001100

R8 = 10011110 01101010 01000101 01100111

E(R(9)-1) = 110011 111100 001101 010100 001000 001010 101100 001111

K₉ = 111000 001101 101011 101011 011011 011111 011101 111111

-----XOR

A9 = 001011 110001 100010 111111 110011 010100 110010 001110

Langkah 5 meng-ekspansi data dari R_{i-1} 32 bit menjadi R_i 48 bit
Sebanyak 16 kali putaran dengan nilai putaran $1 \leq i \leq 16$ menggunakan table Ekspansi

$R_9 = 11010011\ 01000110\ 11110011\ 10011110$

$E(R_{10}-1) = 011010\ 100110\ 101000\ 001101\ 011110\ 100111\ 110011\ 111101$

$K_{10} = 101100\ 011111\ 001101\ 001111\ 111101\ 010101\ 010101\ 010111$

-----XOR

$A_{10} = 110110\ 111001\ 100101\ 001010\ 110000\ 000011\ 101010\ 110010$

$R_{10} = 10101001\ 00001101\ 11110010\ 11001101$

$E(R_{11}-1) = 110101\ 010010\ 100001\ 011011\ 111110\ 100101\ 011001\ 011011$

$K_{11} = 001000\ 010101\ 111111\ 001011\ 101011\ 110011\ 011110\ 000011$

-----XOR

$A_{11} = 111101\ 000111\ 011110\ 001000\ 001001\ 001000\ 010111\ 011101$

$R_{11} = 10001011\ 01010010\ 00011000\ 10001111$

$E(R_{12}-1) = 110001\ 010110\ 101010\ 100100\ 000011\ 110001\ 010001\ 011111$

$K_{12} = 011101\ 010111\ 000111\ 011101\ 100111\ 011011\ 011111\ 011111$

-----XOR

$A_{12} = 101100\ 000001\ 101101\ 010001\ 100110\ 110111\ 001110\ 110110$

$R_{12} = 11000010\ 00111101\ 10001001\ 01100100$

$E(R_{13}-1) = 011000\ 000100\ 000111\ 111011\ 110001\ 010010\ 101100\ 001001$

$K_{13} = 100101\ 111000\ 011111\ 010001\ 111011\ 101001\ 000101\ 000011$

-----XOR

$A_{13} = 111101\ 111000\ 010000\ 101010\ 001111\ 111001\ 000101\ 001000$

$R_{13} = 00101101\ 00111000\ 11011110\ 11100110$

$E(R_{14}-1) = 000101\ 011010\ 100111\ 110001\ 011011\ 111101\ 011100\ 001100$

$K_{14} = 011011\ 111000\ 011110\ 101111\ 101010\ 101111\ 111011\ 111001$

-----XOR

$A_{14} = 010010\ 101110\ 101001\ 000110\ 100111\ 010011\ 000000\ 110110$

Langkah 5 meng-ekspansi data dari R_{i-1} 32 bit menjadi R_i 48 bit
Sebanyak 16 kali putaran dengan nilai putaran $1 \leq i \leq 16$ menggunakan table Ekspansi

$R_{14} = 01101010\ 11101000\ 11100011\ 11000000$

$E(R(15)-1) = 001101\ 010101\ 011101\ 010001\ 011100\ 000111\ 111000\ 000000$

$K_{15} = 101111\ 111001\ 001111\ 101001\ 111111\ 101010\ 101010\ 101111$

-----XOR

$A_{15} = 100010\ 101100\ 011011\ 011100\ 010011\ 010100\ 000100\ 001010$

$R_{15} = 00100001\ 00010010\ 10100011\ 11010000$

$E(R(16)-1) = 000100\ 000010\ 100010\ 100101\ 010100\ 000111\ 111010\ 100000$

$K_{16} = 110010\ 110011\ 101011\ 101001\ 101001\ 101001\ 101001\ 101001$

-----XOR

$A_{16} = 110110\ 110001\ 010100\ 101110\ 010111\ 100110\ 100101\ 010101$

Langkah 6 Setiap vector Ai disubstitusikan ke 8 buah S-box (Substitution box)

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

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	16
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

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

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

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

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

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

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

B1 = 1000 0101 0100 1000 0011 1111 1101 0000

B2 = 0010 0011 0110 1100 0011 0100 1011 1011

B3 = 0101 0010 0001 1001 0100 1101 1000 0001

B4 = 1000 0110 1101 1110 1101 0101 0010 0111

B5 = 0111 1100 0110 0011 0101 1010 0000 1101

B6 = 1011 0000 0111 1001 0000 1100 1000 1001

B7 = 1110 0001 1010 0010 1000 1010 1000 0011

B8 = 0010 1100 0111 0000 1110 0111 0010 1011

B9 = 0010 1011 0110 1110 1111 0011 1111 0001

B10 = 0111 0000 1101 0110 1111 1111 0011 0110

B11 = 0110 0111 1000 0000 0100 1001 1100 1001

B12 = 0010 0011 1000 0100 1011 0111 1101 1101

B13 = 0110 1001 0001 1011 0001 0110 1011 0110

B14 = 1010 0001 0110 0011 0111 0001 0100 1101

B15 = 0001 1101 1011 0100 0000 0011 0010 1111

B16 = 0111 1011 1100 1101 1010 0101 1101 0110

Langkah 7 Setelah dapat Bi, lalu permutasikan bit vector Bi dengan table P-Box

Tabel P-Box

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

Input	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Output	1	0	0	0	0	1	0	1	0	1	0	0	1	0	0	0
Input	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
Output	0	0	1	1	1	1	1	1	1	0	1	0	0	0	0	0

Sehingga hasil yang didapat

$P(B1) = 00110010 \ 10110001 \ 01100000 \ 11011001$

$P(B2) = 01101010 \ 00000011 \ 01011110 \ 11001101$

$P(B3) = 11010100 \ 00000100 \ 10101000 \ 01001011$

$P(B4) = 01100101 \ 11000111 \ 00111101 \ 01111000$

$P(B5) = 10111000 \ 01101101 \ 10001010 \ 00110110$

$P(B6) = 10011100 \ 10000001 \ 00001010 \ 01001111$

$P(B7) = 00010001 \ 11100010 \ 11001011 \ 00000101$

Hasil $P(Bi)$ kemudian di XOR kan dengan $Li-1$ untuk mendapatkan nilai Ri

$L0 : 11111111 \ 00000001 \ 00000100 \ 11111010$

$R0 : 00000000 \ 00000000 \ 00000010 \ 00000001$

$P(B1) = 00110010 \ 10110001 \ 01100000 \ 11011001$

$L(1)-1 = 11111111 \ 00000001 \ 00000100 \ 11111010$

-----XOR

$R1 = 11001101 \ 10110000 \ 01100100 \ 00100011$

$P(B2) = 01101010 \ 00000011 \ 01011110 \ 11001101$

$L(2)-1 = 00000000 \ 00000000 \ 00000010 \ 00000001$

-----XOR

$R2 = 01101010 \ 00000011 \ 01011100 \ 11001100$

$P(B3) = 11010100 \ 00000100 \ 10101000 \ 01001011$

$L(3)-1 = 11001101 \ 10110000 \ 01100100 \ 00100011$

-----XOR

$R3 = 00011001 \ 10110100 \ 11001100 \ 01101000$

$P(B4) = 01100101 \ 11000111 \ 00111101 \ 01111000$

$L(4)-1 \ 01101010 \ 00000011 \ 01011100 \ 11001100$

Langkah 7 Setelah dapat Bi, lalu permutasikan bit vector Bi dengan table P-Box

P(B8) = 00001101 00101111 00101110 10011100

-----XOR

P(B9) = 01100011 01111101 01111110 11000101

R4 = 00001111 11000100 01100001 10110100

P(B10) = 00110111 01100111 10110111 10101010

P(B11) = 01011000 00010100 11101011 00010001

P(B5) = 10111000 01101101 10001010 00110110

P(B12) = 01101011 00110000 01111011 10101001

L(5)-1 = 00011001 10110100 11001100 01101000

P(B13) = 10100110 01101010 11000110 01101001

-----XOR

P(B14) = 10101000 11010101 01101010 10100100

R5 = 10100001 11011001 01000110 01011110

P(B15) = 00001100 00101010 01111101 00110110

P(B16) = 11000011 00011011 11110011 11101011

P(B6) = 10011100 10000001 00001010 01001111

L(6)-1 = 00001111 11000100 01100001 10110100

-----XOR

R6 = 10010011 01000101 01101011 11111011

P(B7) = 00010001 11100010 11001011 00000101

L(7)-1 = 10100001 11011001 01000110 01011110

-----XOR

R7 = 10110000 00111011 10001101 01011011

P(B8) = 00001101 00101111 00101110 10011100

L(8)-1 = 10010011 01000101 01101011 11111011

Langkah 7 Setelah dapat Bi, lalu permutasikan bit vector Bi dengan table P-Box

-----XOR

R8 = 10011110 01101010 01000101 01100111

P(B9) = 01100011 01111101 01111110 11000101

L(9)-1 = 10110000 00111011 10001101 01011011

-----XOR

R9 = 11010011 01000110 11110011 10011110

P(B10) = 00110111 01100111 10110111 10101010

L(10)-1 = 10011110 01101010 01000101 01100111

-----XOR

R10 = 10101001 00001101 11110010 11001101

P(B11) = 01011000 00010100 11101011 00010001

L(11)-1 = 11010011 01000110 11110011 10011110

-----XOR

R11 = 10001011 01010010 00011000 10001111

P(B12) = 01101011 00110000 01111011 10101001

L(12)-1 = 10101001 00001101 11110010 11001101

-----XOR

R12 = 11000010 00111101 10001001 01100100

P(B13) = 10100110 01101010 11000110 01101001

L(13)-1 = 10001011 01010010 00011000 10001111

-----XOR

R13 = 00101101 00111000 11011110 11100110

P(B14) = 10101000 11010101 01101010 10100100

L(14)-1 = 11000010 00111101 10001001 01100100

-----XOR

R14 = 01101010 11101000 11100011 11000000

P(B15) = 00001100 00101010 01111101 00110110

L(15)-1 = 00101101 00111000 11011110 11100110

-----XOR

R15 = 00100001 00010010 10100011 11010000

P(B16) = 11000011 00011011 11110011 11101011

L(16)-1 = 01101010 11101000 11100011 11000000

Langkah 8

Gabungan dari R16 dan L16 lalu permutasikan untuk terakhir kali dengan table inverse initial Permutation (IP-1)

R16 = 10101001 11110011 00010000 00101011

-----XOR

L16 = 00100001 00010010 10100011 11010000

R16 = 10101001 11110011 00010000 00101011

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:

R16L16 = 11011001 00111001 00000000 01000001 00110110 11011001
00010010 01011010

Menghasilkan Output:

Chiper(dalam biner) = **d9 39 00 41 36 d9 12 5a**