Sains dan Komputer (SAINTIKOM)
Vol.18, No.2, Agustus 2019, pp. 161~166
P-ISSN: 1978-6603 E-ISSN : 2615-3475

Implementasi DES (Data Encryption Standard) Untuk Penyandian Data Bill Of Material pada Divisi Produksi PT.Siantar Top, Tbk

Erika Fahmi Ginting, Khairi Ibnutama, Mhd Gilang Suryanata
Program Studi Sistem Informasi, STMIK Triguna Dharma

Article Info
Article history:
Received May 31th, 2019
Revised June 12th, 2019
Accepted Augs 11th, 2019

Keyword:
Bill of Material,
Inventory,
Data Encryption Standard,
Kemanan Data

ABSTRACT
Bill of Material (BOM) memuat data komposisi produk dan juga takaran dalam proses produksi, data ini bisa juga disebut seperti resep. Data ini merupakan salah satu aspek terpenting karena menyangkut rasa dan kualitas produk. Pentingnya data BOM ini biasanya menjadi sasaran bagi para pesaing dunia usaha untuk disalin dan dapat menghasilkan produk sama baiknya ataupun lebih baik lagi. Untuk keamanan data BOM selama ini adalah dengan membatasi akses beberapa pengguna dan hanya pengguna yang berkepentingan saja yang dapat mengakses data ini. Dalam data salinan ini tidak semua material dicantumkan dan beberapa material rahasia di ganti nama menjadi nama unik. Namun keadaan tersebut dianggap kurang efektif mengingat banyak produk yang dihasilkan. Oleh sebab itu, dibuatlah sebuah sistem keamanan data yang memuat informasi BOM tersebut baik data asli maupun data salinan yang dalam pengamanannya memakai algoritma *Data Encryption Standard* (DES) sehingga keamaan data lebih terjaga.

First Author
Nama :Erika Fahmi Ginting
Kantor :STMIK Triguna Dharma
Program Studi :SistemInformasi
E-Mail :erikafg04@gmail.com

1. PENDAHULUAN

Bill of Material (BOM) memuat data komposisi produk dan juga takaran dalam proses produksi. Data ini bisa juga disebut seperti resep yang dalam penyajiannya terdapat beberapa *standard* agar menghasilkan produk yang sesuai dengan keinginan *customer*. Pentingnya data BOM ini biasanya menjadi sasaran bagi para pesaing dunia usaha untuk disalin dan dapat menghasilkan produk sama baiknya ataupun lebih baik lagi. Untuk keamanan data BOM selama ini adalah dengan membatasi akses beberapa pengguna dan hanya pengguna yang berkepentingan saja yang dapat mengakses data ini. Sistem keamanan juga dilakukan dengan menyimpan data yang sebenarnya dan membuat data salinannya. Beberapa salinan data diganti nama menjadi nama unik seperti misalnya STT ZZ 562. Hal itu kurang efektif oleh sebab itu, dibuatlah sebuah sistem keamanan data yang memuat informasi BOM tersebut baik data asli maupun data salinan yang dalam pengamanannya memakai algoritma *Data Encryption Standard* (DES). Kriptografi adalah ilmu mengenai teknik enkripsi dimana data diacak menggunakan suatu kunci enkripsi menjadi sesuatu yang sulit dibaca oleh seseorang yang tidak memiliki kunci dekripsi. DES termasuk dalam kategori Kriptografi *modern* karena berorientasi bit sebab penyandian modern menggunakan media komputer untuk mengolah data.

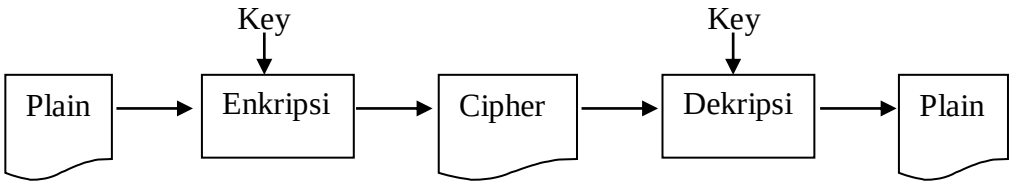
2. METODELOGI PENELITIAN

Penelitian ini dilakukan untuk meningkatkan keamanan data *Bill of Material* (BOM) khususnya pada Divisi Produksi PT. Siantar Top, Tbk dengan memakai algoritma *Data Encryption Standard* (DES).

2.1 Kriptografi

Kriptografi terdiri dari dua kata yaitu "*kryptos*" yang artinya sesuatu yang disembunyikan, rahasia, atau terselubung dan "*graphia*" yang berarti tulisan atau salinan yang dibuat dengan cara atau dengan proses

tertentu. Jadi, kriptografi dapat dijelaskan sebagai ilmu untuk menyembunyikan atau merahasiakan suatu tulisan dengan cara tertentu. Dari penjabaran tersebut, cara untuk menyembunyikan suatu pesan adalah dengan menggunakan suatu algoritma. Adapaun skema dalam Kriptografi seperti berikut ini.



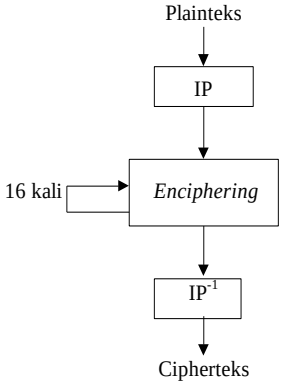
Gambar 2.1 Skema Kriptografi

2.2 Data BOM (*Bill of Material*)

BOM memuat daftar material, sub material, *quantity* yang dibutuhkan untuk mengolah sebuah produk menjadi *finish goods* yang siap jual. BOM biasanya lebih sering digunakan pada perusahaan manufaktur. Secara umum, BOM dapat di sebut sebagai komposisi dalam suatu produk.

2.3 Algoritma Data Encryption Standard (DES)

Algoritma DES termasuk kedalam algoritma simetris karena menggunakan satu buah kunci yang sama dalam mengenkripsi dan mendekripsi data. Algoritma ini tergolong jenis algoritma *block cipher*. Algoritma DES beroperasi pada ukuran blok data sebesar 64 bit. *Plaintext* sebesar 64 bit di enkripsi menjadi *chipertext* sebesar 64 bit dengan menggunakan 56 bit kunci internal yang dibangkitkan dari kunci eksternal sebesar 64 bit. Panjang kunci eksternal 64 bit (sesuai ukuran blok), tetapi hanya 56 bit yang dipakai yaitu 8 bit paritas tidak digunakan. Setiap blok plainteks atau *ciphertext* dienkripsi dalam 16 putaran. Setiap putaran menggunakan kunci internal berbeda.



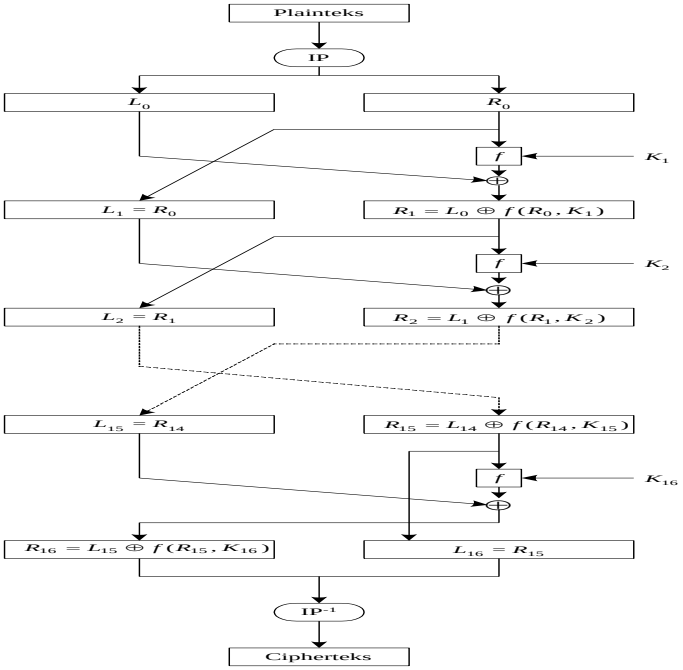
Gambar 2.2 Skema Global Algoritma DES

Ini adalah satu putaran DES. Secara matematis, satu putaran DES dinyatakan sebagai:

$$L_i = R_{i-1}$$

$$R_i = L_{i-1} \oplus f(R_{i-1}, K_i)$$

Dalam fungsi *f* terdapat proses penciptaan kunci internal yang akan dibangkitkan menjadi 16 *sub key* dari masukan kunci eksternal. Kunci – kunci ini juga nantinya akan di X-OR dengan hasil *encode block data*. Proses pengkodean terhadap block data *R* (yang sudah melalui proses pembagian *L* dan *R*) disebut proses *Encode*. Fungsi *f* melibatkan matrik *Permuted Choice*-1 (PC-1), operasi *shift*, dan matrik *Permuted Choice*-2 (PC-2) dalam membangkitkan kunci internal 16 *sub key*. Untuk *encode* blok data, fungsi *f* melibatkan matrik IP, *E-Bit Selection Table*, *S-Boxes*, dan matrik *Permutation* (P).

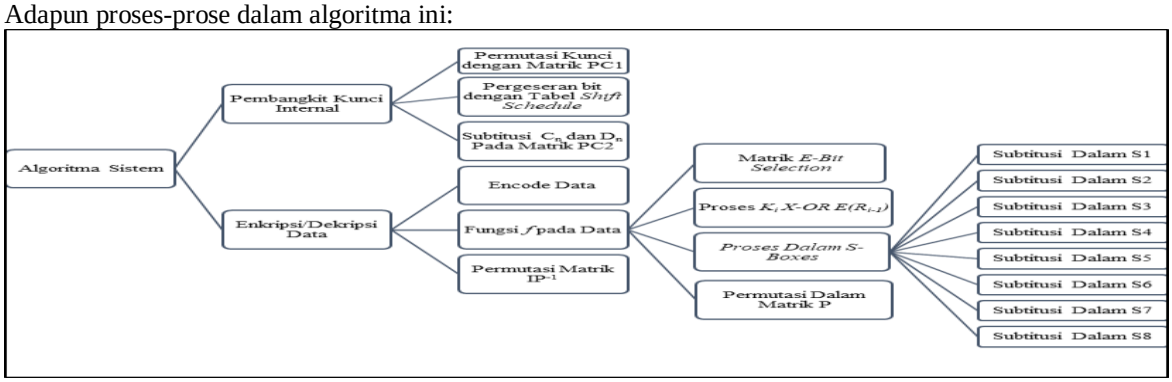


Gambar 2.3 *Flow* Algoritma 3DES 16 Putaran

3. ANALISIS DAN HASIL

Proses enkripsi pada algoritma DES memiliki beberapa tahapan seperti pembangkitan kunci internal, proses *encode* 64 *bit* data, pemrosesan fungsi f hingga pengulangan tahapan – tahapan tersebut sebanyak 16 putaran.

Adapun proses-prose dalam algoritma ini:



Gambar 3.1 Proses – Proses Algoritma DES

3.1 Membuat Pembangkitan Kunci Internal

Proses pembangkitan kunci internal melibatkan matrik PC1 dan PC2 yang akan menciptakan 16 subkunci internal sehingga dapat di proses dalam fungsi f bersama dengan data yang sudah di *encode*.

1. Permutasian Kunci dengan Matrik Permuted Choice 1 (PC1)

Kunci adalah "CUSTOMER" akan konversi menjadi bilangan biner yang dinotasikan sebagai *K*, hasilnya adalah seperti berikut ini.

Kunci-1 = CUSTOMER

Menghasilkan:

K = 01000011010101010101001101010100010011111010011010100010101010010

Kunci asli sebesar 64 *bit* tersebut akan di ubah urutannya mengikuti tabel PC1.

Perubahan urutan *bit – bit* tersebut dinotasikan sebagai *K+*, hasil dari permutasi dengan matrik PC1 di atas akan di bagi menjadi dua yaitu bagian yang akan dinotasikan C_0 dan D_0 masing – masing sebesar 28 *bit* adalah seperti berikut ini.

K+ = 0000000011111111000000001000100101010111101000110000111 0

$\boldsymbol{C_0}$ = 00000000111111110000000001000 $\mathbf{D_0}$ = 10010101011110100011000011110

2. Pergeseran *Bit* dengan Tabel *Shift Schedule* Blok C_n dan D_n terbentuk dari blok C_{n-1} dan D_{n-1} masing – masing untuk n=1,2,…,16 menggunakan *Shift Schedule*, setiap 1 *bit* pertama atau 2 *bit* pertama dari C_{n-1} dan D_{n-1} di geser ke belakang sesuai *Shift Schedule* sehingga *bit* kedua atau ketiga dan selanjutnya akan bergeser kedepan, hasilnya adalah sebagai berikut.

C_1=00000001111111100000000010000 **D_1**=00101010111101000110000111010

C_2=00000011111111000000000100000 **D_2**=01010101111010001100001110100

Sampai (n=1,2,…,16)

C_{16}=00000000111111110000000001000 **D_{16}**=10010101011110100011000011100

3. Subtitusi C_n dan D_n Pada Matrik Permuted Choice 2 (PC2)

Matrik PC2 akan mengubah 56 *bit* C_nD_n menjadi 48 *bit* yang akan dinotasikan sebagai K_n. *bit* ke-9, 18, 22, 25, 35, 38, 43, 54 akan diabaikan. Contoh pertama yang akan di subtitusi adalah C_1D_1 menghasilkan K_1, berikut cara subtitusi C_nD_n terhadap K_n.

C_1 D_1=00000001111111100000000010000001010101111010001100001110 1

C_nD_n terhadap K_n

K_1 = 10110000100100100100000100111100100100011011100 00

Selanjutnya K_2, …, K_{16} di peroleh dari hasil permutasi C_2D_2, …, $C_{16}D_{16}$, cara permutasian setiap C_nD_n sama seperti C_1D_1 sehingga hasil K_2, …, K_{16} adalah sebagai berikut.

K_{16} = 10100000100100100010001010001101000100101100011 1

3.2 Algoritma Enkripsi Data

Setelah proses pembangkitan kunci internal selesai dan menghasilkan 16 sub kunci, selanjutnya proses enkripsi data dengan 16 kali putaran dalam fungsi *f*.

1. *Encode* Data

Kata yang akan disandikan adalah "TIRAMISU", kata tersebut harus di konversi menjadi bilangan bineer, hasil konversi akan dinotasikan sebagai *M*, berikut hasil konversi kata "TIRAMISU".

Berikut hasil dari permutasi *M* yang akan dinotasikan sebagai IP.

IP=11111111100111100111010101010001000000000000000000000001010001000

L_0 = 11111111110001011001000111111010

R_0 = 00000000000000000011001001000100

2. Fungsi *f* pada Data

Pada bagian ini akan dijelaskan bagaimana cara kerja fungsi f yang juga melibatkan blok R_{i-1} dan K_i. Blok R_i sebelum masuk ke fungsi f juga akan diturunkan langsung menjadi L_{i+1}.

a. Matrik *E Bit-Selection*

$E(R_0)$ = 00000000000000000000000000110100100001000001000

$E(R_1)$ = 10000000000101110101011111100101111111000101111 0

b. Proses K_i X-OR $E(R_{i-1})$

Sebelum memasuki tahap proses *S-Boxes*, $E(R_{16})$ dan $E(L_{15})$, …, $E(L_1)$ harus di X-OR terlebih dahulu dengan masing – masing K_i. Contoh penerapan

$\boldsymbol{K_{16}}$ = 10100000100100100010001010001101000100101100011 1

$\boldsymbol{E(R_{16})}$ = 11001111001111110000010101011010011010100000001 1

$\oplus$ = 01101111101011010010011111010111011110001100010 0

c. Proses dalam *S-Boxes* (*S*)

Simulasi kali ini akan menggunakan $E(R_{16}) \oplus$ K16, setiap blok *bit* tersebut dinotasikan sebagai yaitu B_1(011011), B_2(111010), B_3(110100), B_4(100111), B_5(110101), B_6(110111), B_7(100011), B_8(000100). Cara subtitusi B_i kedalam *Si* sama seperti pada proses enkripsi yaitu bit pertama dan terakhir akan dijadikan parameter baris (*i*), sedangkan 4 bit ditengah akan menjadi parameter kolom (*j*).

d. . Permutasi dalam Matrik P

Permutasi P dilakukan terhadap output dari proses *S*-Boxes. Cara permutasi dari matrik P sama seperti pada proses enkripsi. Permutasi P dalam tabel 3.29 adalah dari output *S-Boxes* $E(R_{16}) \oplus K_{16}$ yaitu 0010 1111 0011 1101 1101 1010 1011 1111.

3. Proses Permutasi Matrik Inverse Initial Permutation (IP^{-1})

Sebelum dilakukan permutasian kembali terhadap matrik IP^{-1}, langkah yang harus dilakukan adalah $L_{16} \oplus$ f(R_{16}, K_{16}). Untuk selanjutnya, R_{15} langsung diturunkan dari R_{16}, R_{14} langsung diturunkan dari L_{15} dan seterusnya, hasilnya adalah sebagai berikut.

L_{16} = 01010000011100101000011001110111
$P(E(R_{16}) \oplus K_{16})$ = 01001010011000001111010000001111
$\oplus$ = 00011010000100100111001001111000 (akan menjadi L_{15})

Setelah enam belas putaran selesai di proses, maka akan didapat L_1 dan R_1, Hasil L_0 dan R_0 disubtitusi di matrik IP^{-1}.

L_0R_0 = 11111111 11000101 10010001 11111010 00000000 00000000 00110010 01000100

Tabel 3.30 Permutasi L_0R_0 dalam Matrik IP^{-1}

40	0	8	1	48	0	16	1	56	0	24	1	64	0	32	0
39	0	7	1	47	0	15	0	55	1	23	0	63	0	31	1
38	0	6	1	46	0	14	1	54	0	22	0	62	1	30	0
37	0	5	1	45	0	13	0	53	0	21	0	61	0	29	1
36	0	4	1	44	0	12	0	52	1	20	1	60	0	28	1
35	0	3	1	43	0	11	0	51	1	19	0	59	0	27	1
34	0	2	1	42	0	10	1	50	0	18	0	58	1	26	1
33	0	1	1	41	0	9	1	49	0	17	1	57	0	25	1

IP^{-1} = 0101 0100 0100 1001 0101 0010 0100 0001 0100 1101 0100 1001 0101 0011 0101 0101

Hex = 544952414D495355

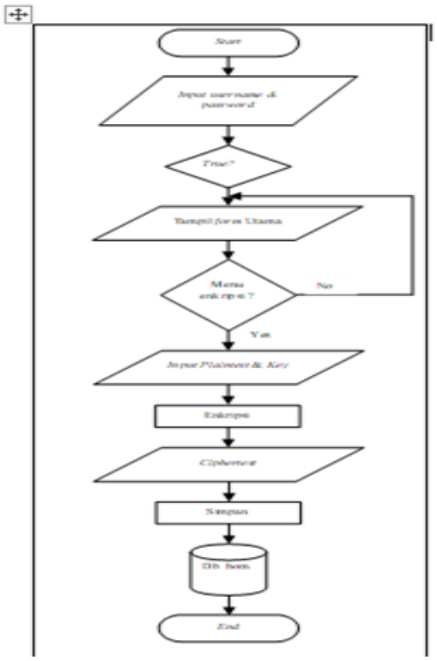
Char = “TIRAMISU”

Dari proses enkripsi yang dilakukan terhadap *plaintext* “TIRAMISU” menggunakan kunci “CUSTOMER” menghasilkan *ciphertext* “‹=€Ù9s¬” kemudian proses dekripsi dari *ciphertext* “‹=€Ù9s¬” menggunakan kunci “CUSTOMER” menghasilkan *plaintext* “TIRAMISU”.

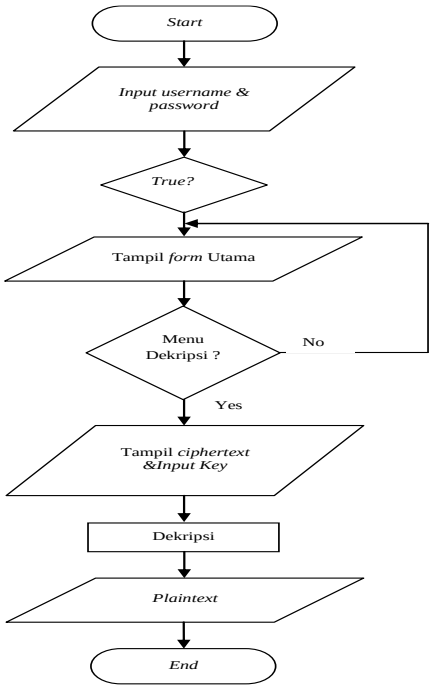
Pada dasarnya algoritma dekripsi dari DES sama seperti proses enkripsi, hanya saja matrik IP tetap digunakan diawal permutasi untulk mengatur kembali bit – bit yang telah disebar dari matrik IP^{-1}.

3.3 Flowchart

*D*imulai dari *input plaintext* hingga menjadi *ciphertext* dan proses dekripsi *ciphertext* menjadi *plaintext.*



Gambar 3.2 *Flowchart* Enkripsi



Gambar 3.3 *Flowchart* Deskripsi

4. KESIMPULAN

Dari hasil penelitian yang telah dilakukan diperoleh kesimpulan yaitu : Implementasi DES (Data Encryption Standard) Untuk Penyandian Data Bill Of Material pada Divisi Produksi PT.Siantar Top, Tbk diperoleh data BOM dapat diamankan dengan mengimplementasikan algoritma DES dimana data BOM dienkripsi per material dalam setiap BOM dalam tahapannya Algoritma DES mengenkripsi dengan memproses data dalam jaringan feistel sebanyak 16 kali putaran. Aplikasi hanya bisa mengenkripsi data dalam bentuk teks serta menggunakan kunci yang simetris dan dapat digunkan untuk mengamankan data pada BOM pada Divisi Produksi untuk mempertahakan kualitas produk sebuah perusahaan sehingga tidak diketahui oleh ppara pesaing

REFERENSI

[1].Basri. 2016."Kriptografi Simetris Dan Asimetris Dalam Perspektif Keamanan Data Dan Kompleksitas Komputasi". *Ilmiah Ilmu Komputer*, *2*(2), 18-20.

[2] Kak, A. 2017. "Computer and Network Security". West Lafayette, Indiana: Purdue University

[3] Mandal, P. C. 2012. Evaluation of performance of the Symmetric Key Algorithms: DES, 3DES ,AES and Blowfish. *Journal of Global Research in Computer Science*. *3*(8). 67.

[4] Patrick, K. T. 2010. The Data Encryption Standard Thirty Four Years Later: An Overview. *African Journal of Mathematics and Computer Science Research*, *3*(10). 267.

[5] Primatha, R. 2011. Penerapan Enkripsi Dan Dekripsi File Menggunakan Algoritma Data Encryption Standard (DES). *Jurnal Sistem Informasi*, *3*(2), 373-374.