METODE ALGORITMA DES UNTUK KEAMANAN DATA

Michael Tjoanda, Rendy Saputra, Alten Cornelius
Universitas Katolik Musi Charitas
Email: michaeltjoanda12345@gmail.com

ABSTRAK

Penelitian ini mendalami keamanan data dengan fokus pada Algoritma Data Encryption Standard (DES) dalam lingkungan digital. DES, meskipun pernah dianggap standar, telah menghadapi tantangan dari kemajuan teknologi. Metode analisis digunakan untuk mengevaluasi kekuatan kriptografi, kompleksitas komputasi, dan resistensi terhadap serangan DES. Proses enkripsi, termasuk permutasi awal, pembangkitan kunci, dan 16 putaran enkripsi, diuraikan dengan contoh penerapan pada teks "COMPUTER." Hasil penelitian menyajikan pemahaman mendalam terhadap kualitas dan keamanan DES. Meskipun masih digunakan, perlu pertimbangan untuk beralih ke standar keamanan data modern seperti Advanced Encryption Standard (AES) yang lebih kuat.

Kata Kunci : Algoritma DES, Keamanan Data, Kriptografi.

ABSTRACT

This research delves into data security with a focus on the Data Encryption Standard (DES) algorithm in the digital environment. DES, once considered a standard, has faced challenges due to technological advancements. Analytical methods are employed to evaluate the cryptographic strength, computational complexity, and resistance to DES attacks. The encryption process, including initial permutation, key generation, and 16 encryption rounds, is elucidated with an application example using the text "COMPUTER." The research findings provide an in-depth understanding of the quality and security of DES. Although still in use, consideration should be given to transitioning to modern data security standards such as the Advanced Encryption Standard (AES), known for its enhanced strength.

Keywords ***:*** *DES algorithm, Data security, Cryptography*

1. PENDAHULUAN

Di dunia digital saat ini, keamanan data merupakan hal yang sangat penting. Volume dan kompleksitas data yang disimpan dan dikirim telah berkembang seiring dengan kemajuan teknologi digital. Selain itu, hal ini juga meningkatkan kemungkinan terjadinya pembobolan data yang dapat menyebabkan kerugian besar.

Algoritma Data Encryption Standard (DES) adalah salah satu teknik kriptografi yang sering digunakan untuk melindungi data. Dalam berbagai skenario, DES telah terbukti efektif dalam menjaga integritas dan kerahasiaan data. Akan tetapi, DES juga memiliki beberapa kekurangan yang membuatnya terbuka terhadap serangan cryptoanalytic yang lebih canggih.

Penelitian ini bermaksud untuk menguji keamanan DES terhadap serangan cryptoanalytic yang semakin kompleks dengan mempertimbangkan isu-isu tersebut. Agar keamanan DES dapat terus efektif dalam mengamankan data di era digital, penelitian ini juga berusaha memberikan saran untuk perbaikan.

Di dunia digital saat ini, keamanan data merupakan hal yang sangat penting. Volume dan kompleksitas data yang disimpan dan dikirim telah berkembang seiring dengan kemajuan teknologi digital. Selain itu, hal ini juga meningkatkan kemungkinan terjadinya pembobolan data yang dapat menyebabkan kerugian besar.

Algoritma Data Encryption Standard (DES) adalah salah satu teknik kriptografi yang sering digunakan untuk melindungi data. Dalam berbagai skenario, DES telah terbukti efektif dalam menjaga integritas dan kerahasiaan data. Akan tetapi, DES juga memiliki beberapa kekurangan yang membuatnya terbuka terhadap serangan cryptoanalytic yang lebih canggih.

Penelitian ini bermaksud untuk menguji keamanan DES terhadap serangan cryptoanalytic yang semakin kompleks dengan mempertimbangkan isu-isu tersebut. Agar keamanan DES dapat terus efektif dalam mengamankan data di era digital, penelitian ini juga berusaha memberikan saran untuk perbaikan.

Sebelum memulai penelitian, diperlukan tinjauan terhadap literatur yang ada. Banyak penelitian sebelumnya mengenai keamanan DES telah dilakukan baik di dalam negeri maupun di luar negeri. Penelitian sebelumnya oleh Ariska dan Wahyuddin (2022) dilakukan di Indonesia. Dalam penelitian ini, keamanan DES terhadap serangan cryptoanalytic yang semakin kompleks diperiksa. Temuannya menunjukkan bahwa meskipun DES relatif aman dari serangan kriptanalisis diferensial, DES rentan terhadap serangan kriptanalisis yang lebih canggih. Penelitian sebelumnya yang dilakukan di luar negeri adalah penelitian oleh Ir.Rinaldi Munir,M.T. (2004). Penelitian ini menganalisa keamanan DES terhadap serangan kriptanalisis linier. Hasil penelitian menunjukkan bahwa DES rentan terhadap serangan kriptanalisis linier.

Berdasarkan tinjauan pustaka dari penelitian sebelumnya, dapat disimpulkan bahwa DES memiliki beberapa kelemahan yang membuatnya rentan terhadap serangan kriptoanalisis yang lebih canggih. Oleh karena itu, diperlukan penelitian lebih lanjut untuk meningkatkan keamanan DES agar tetap efektif dalam melindungi data di era digital.

2. METODE

Pada bagian ini, kami akan memberikan penjelasan mendalam mengenai metode yang kami terapkan dalam penelitian ini. Metode analisis yang digunakan dalam penelitian ini adalah metode penelitian kuantitatif. Fokus utama dari metode ini adalah untuk melakukan analisis mendalam terhadap kinerja perhitungan Algoritma DES dalam beberapa aspek kunci, termasuk:

- Kekuatan kriptografi: Ketahanan Algoritma DES terhadap serangan-serangan kriptografi, seperti serangan brute-force, serangan differential, dan serangan linear.

- Kompleksitas komputasi: Kompleksitas komputasi yang dibutuhkan untuk

menjalankan Algoritma DES, baik untuk proses enkripsi maupun dekripsi.

- Resistensi terhadap serangan: Ketahanan Algoritma DES terhadap serangan-serangan fisik, seperti serangan man-in-the-middle dan serangan side-channel.

Untuk melakukan analisis terhadap aspek-aspek tersebut, kami akan menggunakan metode-metode berikut:

- Analisis teoretis: Analisis teoretis akan dilakukan untuk mempelajari prinsip kerja Algoritma DES dan untuk memahami kelemahan-kelemahan potensial dari algoritma tersebut.

- Eksperimen: Eksperimen akan dilakukan untuk mengukur kinerja Algoritma DES secara empiris.

2.1 Prosedur Kinerja Algoritma DES

Algoritma DES adalah algoritma enkripsi simetris yang menggunakan kunci 56-bit untuk mengenkripsi dan mendekripsi data. Algoritma ini terdiri dari 16 putaran, masing-masing putaran terdiri dari dua langkah:

- Permutasi awal: Permutasi awal digunakan untuk mengacak data input.
- Fungsi Feistel: Fungsi Feistel adalah fungsi kunci yang digunakan untuk melakukan enkripsi atau dekripsi data.

Pada proses enkripsi, data input akan diproses melalui 16 putaran fungsi Feistel, dengan kunci yang berbeda pada setiap putaran. Data output dari putaran ke-i akan menjadi data input untuk putaran ke-(i+1).

Pada proses dekripsi, data input akan diproses melalui 16 putaran fungsi Feistel, dengan kunci yang sama dengan urutan kunci yang digunakan pada proses enkripsi. Data output dari putaran ke-i akan menjadi data input untuk putaran ke-(i-1).

2.2 Tahap-tahap Perhitungan dari Ciphertext ke Dekripsi

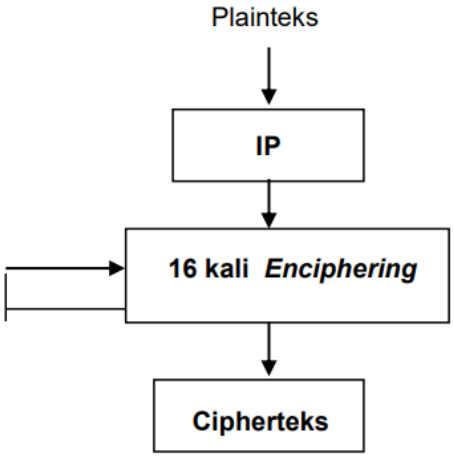
Pada proses dekripsi, data input (ciphertext) akan diproses melalui 16 putaran fungsi Feistel, dengan kunci yang sama dengan urutan kunci yang digunakan pada proses enkripsi. Data output dari putaran ke-i akan menjadi data input untuk putaran ke-(i-1).

Secara umum, tahap-tahap perhitungan dari ciphertext ke dekripsi adalah sebagai berikut:

- Initilisasi: Inisialisasi variabel-variabel yang dibutuhkan untuk proses dekripsi.
- Permutasi awal: Permutasi awal digunakan untuk mengacak data input.
- Fungsi Feistel: Fungsi Feistel digunakan untuk melakukan dekripsi data.
- Inverse IP: Inverse IP digunakan untuk mengembalikan data output ke bentuk aslinya.

Skema global algoritme DES terlihat seperti ini (lihat Gambar 1):

1. Matriks permutasi awal (IP) digunakan untuk melakukan permutasi pada blok plainteks.
2. Setelah itu, hasil mutasi pertama diperoleh sekitar 16 kali (16 putaran). Ada kunci internal yang berbeda untuk setiap putaran.
3. Selanjutnya, invers matriks permutasi awal (IP-1) digunakan untuk mengubah hasil dekripsi menjadi blok kriptografi.

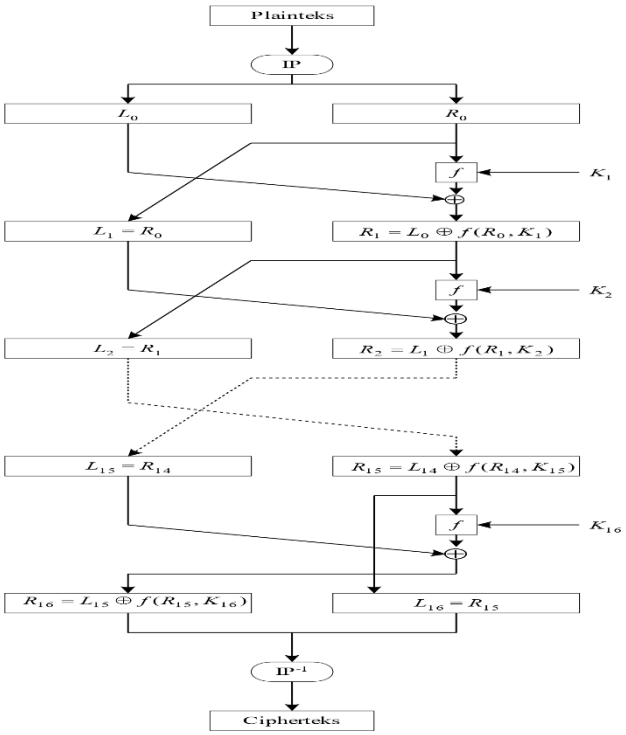


Gambar 1. Skema Global DES

Blok plaintext dibagi menjadi dua bagian, kiri (L) dan kanan (R), yang masing-masing memiliki panjang 32-bit selama proses penulisan. Kedua komponen ini digunakan dalam 16 putaran DES. Blok R berfungsi sebagai input untuk fungsi transformasi, yang dilambangkan sebagai f, pada setiap putaran I. Kunci internal Ki dan blok R dilekatkan pada fungsi f. Untuk membuat blok R yang baru, output dari fungsi f di-XOR-kan dengan blok L. Sebaliknya, blok R yang sebelumnya berfungsi sebagai sumber yang tidak bergerak untuk blok L yang baru dibuat.

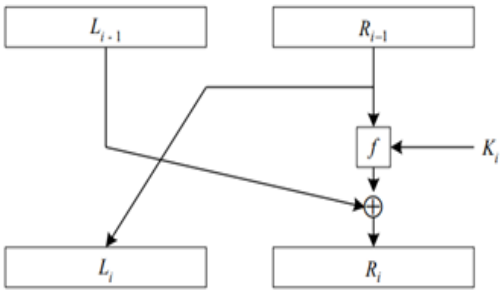
Deskripsi sistematis dari putaran DES pertama adalah sebagai berikut:

- Ri - 1 = Li
- Ri - 1 Li = f (Ri - 1, Ki)



Gambar 2. Algoritma Enkripsi dengan DES

Skema yang lebih komprehensif dari algoritma DES dapat ditemukan pada Gambar 3. Satu putaran DES digunakan untuk memodelkan jaringan serat optik yang diberikan (lihat Gambar 3).



Gambar 3. Jaringan Feistel untuk sebuah DES tunggal

Perhatikan Gambar 2, yang menyatakan bahwa pre-ciphertext dari penyandian ini adalah (R16, L16) jika (L16, R16) adalah output dari putaran ke-16. Modifikasi pertama dari keseimbangan IP-1 sehubungan dengan blok cipherteks menghasilkan cipherteks yang jelas.

2.3 Permutasi Awal

Mutasi pertama (IP) diterapkan pada blok plaintext sebelum tahap pertama selesai. Mutasi pertama bertujuan untuk memecah plaintext sehingga setiap bit di dalamnya memiliki karakter yang unik. Untuk menyelesaikan transaksi, gunakan matriks permutasi awal yang disediakan:

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

Gambar 4 Matriks Permutasi Awal

Memahami tabel/matriks: "Pindahkan bit ke-58 ke bit 1" dan "Pindahkan bit ke-50 ke bit 2" adalah arti dari dua entri kiri teratas (58 dan 50), masing-masing.

2.4 Pembangkitan Kunci Internal

Karena adanya enam belas putaran, enum belas kunci internal, yaitu K, Kz,..., K16 -, diperlukan. Kode internal ini dapat dibuat bersamaan dengan penyandian atau rancangan sebelumnya. Kunci eksternal yang dapat diakses oleh pengguna digunakan untuk menghasilkan kunci internal. Sebuah set karakter 64-bit

digunakan untuk merepresentasikan kunci eksternal.

Anggap saja K adalah kunci eksternal yang terdiri dari 64 bit. Memanfaatkan matriks PC-1 untuk permutasi, kunci eksternal digunakan sebagai input untuk permutasi dengan cara yang dijelaskan di bawah ini:

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

Gambar 5. Matriks Permutasi kompresi PC-1

Pada mutasi ini, setiap bit kedelapan (bit paritas) dari delapan byte kunci dihilangkan. Dapat disimpulkan bahwa kunci DES memiliki panjang 56 bit karena hasil dari mutasi tujuh kali adalah 56 bit. Sebagai contoh, 56 bit tersebut dibagi menjadi dua segmen yang masing-masing terdiri dari 28 bit, kiri dan kanan, yang sebagian besar diwakili oleh CO dan DO:

CO: berisi sejumlah kecil informasi dari K pada posisi

57,49,41,33,25,17,9,1,58,50,42,34,26,18

DO: berisi sejumlah kecil informasi dari K pada posisi

10, 3, 60, 52, 44, 36, 27, 19, 51, 43, 35, dan seterusnya

Lakukan: letakkan informasi bit demi bit dari K di tempat yang telah ditentukan.

15, 7, 62, 54, 46, 38, 30, 22, 63, 55, 47, 39, 31, 23,

13, 5, 28, 20, 12, 4, 61, 53, 45, 37, 29, 21, dan seterusnya.

Terakhir, kedua sisi diputar ke kiri (left shift) selama satu atau dua derajad setiap putaran.

3. HASIL DAN PEMBAHASAN

Enkripsi data algoritma DES (Data Encryption System):

Plaintext (x) = COMPUTER
Key (k) = 1A 3C 5E 7F 92 B4 D6 F8

1. Langkah pertama:
   Ubah plaintext dan key menjadi format biner.

TABEL I. KONVERSI BINER

Plaintext	Biner	Key	Biner
C	01000011	1A	00011010
O	01001111	3C	00111100
M	01001101	5E	01011110
P	01010000	7F	01111111
U	01010101	92	10010010
T	01010100	B4	10110100
E	01000101	D6	11010110
R	01010010	F8	11111000

2. Langkah kedua:
   Permutasi Awal (IP) pada bit plaintext menggunakan tabel IP.

TABEL II. INITIAL PERMUTATION

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 yang terdapat pada plaintext ke-58 dikorelasikan dengan Posisi 1, Posisi 2 dengan urutan bit yang terdapat pada plaintext ke-50, Posisi 3 dengan urutan bit yang terdapat pada plaintext ke-42, dan seterusnya. Jadi, hasil keluarannya adalah:

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

L0 : 11111111 10111000 01110110 01010111
R0 : 00000000 00000000 00000110 10000011

3. Langkah ketiga:

Memanfaatkan tabel permutasi kompresi PC-1, buat kunci yang akan digunakan untuk mengenkripsi teks biasa. Pada langkah ini, kompresi dicapai dengan mengurangi setiap blok kunci dari 64 bit menjadi 56 bit sebanyak satu bit.

CD(k): 0001100 0011100 0101110 0111111 1001001 1011010 1101100 1111000
C0 : 0001100 0011100 0101110 0111111
D0 : 1001001 1011010 1101100 1111000

Pecah CD (k) menjadi dua bagian kiri dan kanan, sehingga menjadi
C0 : 0001100 0011100 0101110 0111111
D0 : 1001001 1011010 1101100 1111000

4. Langkah keempat:

Dengan menggunakan waktu rotasi yang tercantum dalam tabel rotasi sebagai panduan, geser ke kiri (shift kiri) pada C0 dan D0 satu atau dua kali. Pindahkan sedikit ke kiri untuk putaran pertama. Pindahkan satu bit ke kiri untuk putaran kedua. Pindahkan dua bit ke kiri untuk putaran ketiga, dan seterusnya. Hasil keluarannya adalah sebagai berikut:

C0 : 0001100 0011100 0101110 0111111
D0 : 1001001 1011010 1101100 1111000

Putaran 1:
C1 : 0011000 0111000 1011100 1111110 (1 kali shift ke kiri)
D1 : 0010011 0110101 1011011 1110001 (1 kali shift ke kiri)

Putaran 2:
C2 : 0110001 1100001 0111001 1111100 (2 kali shift ke kiri)
D2 : 0100110 1101011 0110111 1100011 (2 kali shift ke kiri)

Putaran 3:
C3 : 1000110 0000110 1110011 1111000 (2 kali shift ke kiri)
D3 : 1001101 1011101 1101111 1000110 (2 kali shift ke kiri)

Putaran 4:
C4 : 0001101 1001101 0110111 1111000 (2 kali shift ke kiri)
D4 : 0101100 1101110 1110110 0001101 (2 kali shift ke kiri)

Putaran 5:
C5 : 0110011 0101100 1111111 1000110 (2 kali shift ke kiri)
D5 : 0001001 1100111 1011000 1010101 (2 kali shift ke kiri)

Putaran 6:
C6 : 0010011 1010001 1111110 0001100 (2 kali shift ke kiri)
D6 : 0111000 1110100 1101010 1101101 (2 kali shift ke kiri)

Putaran 7:
C7 : 0100101 0100100 0111000 0011001 (2 kali shift ke kiri)
D7 : 1110001 0011101 0110101 0110010 (2 kali shift ke kiri)

Putaran 8:
C8 : 1001010 1001000 1110000 1100110 (2 kali shift ke kiri)
D8 : 0110111 0110100 1010100 0101111 (2 kali shift ke kiri)

Putaran 9:
C9 : 0101010 1011111 1110000 1100110 (2 kali shift ke kiri)
D9 : 0011110 0011110 1010101 0110011 (2 kali shift ke kiri)

Putaran 10:
C10: 0101010 1111111 1000011 0011001 (2 kali shift ke kiri)
D10: 1111000 1111010 1010101 1001100 (2 kali shift ke kiri)

Putaran 11:
C11: 0101011 1111110 0001100 1100101 (2 kali shift ke kiri)
D11: 1100011 1101010 1010110 0110011 (2 kali shift ke kiri)

Putaran 12:
C12: 0101111 1111000 0110011 0010101 (2 kali shift ke kiri)

D12: 0001111 0101010 1011001 1001111 (2 kali shift ke kiri)

Putaran 13:
C13: 0111111 1100001 1001100 1010101 (2 kali shift ke kiri)
D13: 0111101 0101010 1100110 0111100 (2 kali shift ke kiri)

Putaran 14:
C14: 1111111 0000110 0110010 1010101 (2 kali shift ke kiri)
D14: 1110101 0101011 0011001 1110001 (2 kali shift ke kiri)

Putaran 15:
C15: 1111100 0011001 1001010 1010111 (2 kali shift ke kiri)
D15: 1010101 0101100 1100111 1000111 (2 kali shift ke kiri)

Putaran 16:
C16: 1111000 0110011 0010101 0101111 (2 kali shift ke kiri)
D16: 0101010 1011001 1001111 0001111 (2 kali shift ke kiri)

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 III.
PERMUTATION COMPRESSION 2 (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

Kunci ronde pertama = 10011000 11010101 10101101 01001001 11101010 01100010 11111101 00001111...
(Lakukan langkah yang sama sebanyak 16 kali untuk mendapatkan kunci ronde 1-16)

5. Langkah kelima:

Pada langkah ini, dengan menggunakan Tabel Ekspansi (E), perluas data Ri-1 32 bit menjadi Ri 48 bit sebanyak 16 kali dengan nilai rotasi 1<= i <=16.

TABEL IV. EKSPANSI (E)

32	1	2	3	4	5
4	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

R16: 01101110 10100010 10101000 10110001
E(R16): 01000110 11011010 10001011 10000101 01010110 10100100 10101000 10110001

6. Langkah keenam:

Untuk menghasilkan vektor keluaran Bi 32, setiap vektor Ai disubstitusikan ke dalam delapan S-Box (Substitution Box). Blok pertama diganti dengan S1, blok kedua diganti dengan S2, dan seterusnya.

TABEL V. INITIAL PERMUTATION-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	11	49	17	57	25

Sehingga Input:
R16L16 = 01000000 11011011 00101111 01101010 11010010 01110000 01100101 01011110

Menghasilkan Output:
Cipher (dalam biner) = 01000000 11011011 00101111 01101010 11010010 01110000 01100101 01011110

Cipher (dalam hexa) = **40 DB 2F 6A D2 70 65 5E**

Dalam contoh aplikasi Data Encryption Standard (DES) ini, kunci '1A 3C 5E 7F 92 B4 D6 F8' digunakan untuk mengubah kata 'COMPUTER' menjadi ciphertext. Untuk

memulainya, setiap karakter dalam plaintext diubah menjadi bentuk biner, dan kunci juga diubah menjadi format biner. Plaintext diubah menjadi ciphertext melalui serangkaian putaran dengan menggunakan fungsi-fungsi unik seperti substitusi dan permutasi.

Hasil akhirnya adalah ciphertext dalam format heksadesimal (40 DB 2F 6A D2 70 65 5E) dan biner (01000000 11011011 00101111 01101010 11010010 01110000 01011110). Algoritma DES mampu menghasilkan data yang aman dan sulit dipecahkan dengan menggunakan kunci '1A 3C 5E 7F 92 B4 D6 F8'.

Dengan mempertimbangkan implementasi DES secara hati-hati, kita dapat meningkatkan keamanan data dengan menggunakan metode enkripsi yang kuat. Dengan melakukan proses deskripsi pada ciphertext yang dihasilkan menggunakan kalkulator perhitungan, kita dapat mengkonfirmasi keakuratan enkripsi. Dengan menggunakan kunci yang sama, ciphertext diubah kembali menjadi plaintext asli dalam langkah-langkah ini. Jika beruntung, output dari deskripsi ini akan sama persis dengan plaintext asli yang diberikan sebagai input pertama.

Prosedur-prosedur ini memungkinkan untuk pengujian dan verifikasi keamanan algoritma DES secara menyeluruh, menjamin bahwa operasi enkripsi dan deskripsi beroperasi sesuai dengan yang diharapkan dan memberikan hasil yang dapat diandalkan. Dasar yang kuat untuk keamanan informasi akan disediakan oleh pemeliharaan kerahasiaan data algoritma DES yang efektif, yang akan dijamin dengan implementasi yang hati-hati dan pengujian yang komprehensif.

4. PENUTUP

4.1. Kesimpulan

Berdasarkan penelitian yang telah dilakukan sebelumnya, kesimpulan yang dapat diambil adalah sebagai berikut:

1. Algoritma kriptografi kunci simetris yang populer untuk enkripsi data yang menjaga integritas dan kerahasiaan data adalah Algoritma Data Encryption Standard (DES).

2. Kekuatan kunci yang digunakan menentukan seberapa aman algoritma DES. Namun demikian, DES sekarang rentan terhadap serangan karena kemajuan dalam kriptanalisis dan terobosan komputasi.

3. Penelitian ini menguji seberapa baik kinerja algoritma DES ketika mengenkripsi plaintext menjadi ciphertext dengan menggunakan berbagai macam teknik, termasuk pembuatan kunci internal, permutasi awal, kompresi permutasi dan fungsi pengganti.

4. Dengan menggunakan kunci 13 34 57 79 9B BC DF F1, plaintext "KOMPUTER" dienkripsi untuk menghasilkan ciphertext dalam format heksadesimal 56 f1 d5 c8 52 af.

5. Meskipun memiliki sejarah panjang dalam hal keefektifannya, algoritma DES telah digantikan oleh AES (Advanced Encryption Standard) yang lebih tangguh dalam menghadapi ancaman serangan kriptanalisis modern.

4.2. Saran

1. Diperlukan lebih banyak investigasi terhadap kelemahan dalam algoritma DES. Sangatlah penting untuk memahami sejauh mana algoritma ini tetap aman untuk digunakan pada masa sekarang. Juga perlu untuk menganalisa serangan cryptoanalysis terbaru terhadap DES.

2. Perbandingan kinerja yang menyeluruh terhadap keamanan, kecepatan, dan efisiensi komputasi dari algoritma DES dan AES harus dilakukan. Hasil perbandingan tersebut dapat digunakan sebagai panduan untuk memutuskan algoritma mana yang paling cocok untuk digunakan dalam praktek.

3. Studi kasus penggunaan algoritma DES pada program atau sistem tertentu dapat dilakukan. Misalnya, dalam jaringan nirkabel, aplikasi mobile, sistem

database, dan lain sebagainya. Wawasan yang lebih berguna akan diperoleh dari studi kasus ini.

4. Cara lain untuk menangani masalah keamanan data yang akan datang adalah dengan membuat algoritma kriptografi baru yang lebih kuat dan efektif daripada DES atau AES. Sangatlah bermanfaat untuk menyelidiki dan meneliti lebih lanjut di bidang ini.

5. Penting untuk mempromosikan pendidikan publik dan sosialisasi tentang nilai keamanan data dan privasi. Sebagai hasilnya, lebih banyak orang akan sadar akan kebutuhan untuk menggunakan prosedur keamanan data.

5. DAFTAR PUSTAKA

Adriansyah, T. (2021) 'Implementasi Kombinasi Algoritma RSA dan Algoritma DES Pada Aplikasi Pengaman Pesan Teks', Jurnal Sains Manajemen Informatika dan Komputer, 20(1), pp. 38–43. Available at: https://ojs.trigunadharma.ac.id/.

Annisa, T. and Ina, T. ,Siregar (2021) 'Implementasi Kriptografi untuk Keamanan Data dan Jaringan menggunakan Algoritma DES', JURTI, 5(1).

Ariska and wahyuddin (2022) Penerapan kriptografi menggunakan algoritma Data Encryption Standard (DES). Available at: https://jurnal.umpar.ac.id/index.php/sylogh ttps://jurnal.umpar.ac.id/index.php/sylog.

Fachri, B. and Roy, M. ,Sembiring (2020) 'Pengamanan Data Teks Menggunakan Algoritma DES Berbasis Android', Jurnal Media Informatika Budidarma, 4(1), p. 110. Available at: https://doi.org/10.30865/mib.v4i1.1700.

Maria, S. and Jonson, M. (2021) 'Perancangan Aplikasi Penyandian Teks Menggunakan Algoritma Triple DES', 3(3), pp. 197–201.

Muhamad, D. (2011) Analisa Proses Enkripsi dan Deskripsi dengan Metode DES.

Muklas, A. ,Putra et al. (2022) 'Perancangan Aplikasi Enkripsi & Deskripsi pada Dokumen Dengan Algoritma Triple DES Berbasis Web'.

Novelius, B. and Anita, S. (2020) 'Analisis dan Perancangan Keamanan Data Teks Menggunakan Algoritma Kriptografi DES (Data Encryption Standard)', Jurnal Teknologi Informasi, 3.

Sabar, H. (2018) 'Implementasi enkripsi dalam pengamanan file data karyawan dengan metode algoritma DES (Data Encryption Standard)'.

Widiarti, R. ,Maya, Azanuddin and Elfitriani (2022) 'Implementasi Kriptografi Pengamanan Data Nilai Siswa Menggunakan Algoritma DES', Jurnal Sains Manajemen Informatika dan Komputer, 21(1), pp. 1–9. Available at: https://ojs.trigunadharma.ac.id/index.php/jis.