SISTEM APLIKASI UNTUK KEAMANAN DATA DENGAN ALGORITMA 'DES' (*Data Encryption Standard*)

Dadang Priyanto[1], Raisul Azhar[2]
[1,2]Dosen, STMIK Bumigora Mataram
[1]dadang_tesis@yahoo.com, [2]raisulazhar@yahoo.co.id

Abstract

Data is very important its existence to be protected if the data has a value, or a specific confidentiality. There are so many techniques that can be used to secure the data, one is to apply the method Encryption and Decryption. In the Encryption Method and Description using various types of algorithms. Applications built using algorithms Encription Data Standard (DES). This algorithm emphasizes on input (plaintext) whose length is 64 bits and a key length of 56 bits, and through 16 stages or iterations. Results from making this application can be used to encrypt all files except the video. This application has an encryption rate of 3 sec / each file capacity of 1024 bytes. The rate of speed encryption and decryption with polygons forming linear graph is a straight line.

Key word : Encryption, Description, Iterations, Data

I. PENDAHULUAN

Data merupakan salah satu bagian yang sangat penting dalam dunia informatika. Semakin tinggi nilai suatu data (dalam arti nilai item data yang dikandungnya), semakin tinggi pula resiko akan kerusakan atau kehilangan data tersebut.[1] Hal ini dapat dimaklumi, karena data yang bernilai tinggi ibarat "*nyawa*" bagi manusia, sehingga bila suatu data sudah mengalami kerusakan atau bahkan hilang, maka data tersebut sudah tidak ada artinya lagi bagi informasi. Sebagian pelaku kejahatan yang umum terjadi dapat dilihat pada tabel 1.

Tabel 1. Beberapa Orang yang Menyebabkan Masalah Keamanan dan Alasannya (Tanenbaum, 1997).[1]

Musuh	Tujuan
Mahasiswa	Sekedar iseng mengintip *email* orang lain
Hacker	Menguji sistem keamanan orang lain; mencuri data
Sales Rep.	Mengaku mewakili seluruh daratan Eropa, tidak hanya Andorra
Businessman	Menemukan rencana pemasaran yang strategis perusahaan saingan
Bekas Pegawai	Membalas dendam karena telah dipecat
Akuntan	Menggelapkan uang perusahaan
Pialang Saham	Menyangkal janji yang telah dibuat bagi pelanggan dengan memakai *email*
Penipu	Mencuri nomor kartu kredit untuk belanja
Mata-mata	Mempelajari kekuatan militer musuh
Teroris	Mencuri rahasia senjata kuman

Dari daftar tabel 1 bahwa jaringan yang aman meliputi lebih dari sekedar menjaga jaringan agar bebas dari error. Melainkan hal itu juga menyangkut musuh yang cerdas, berdedikasi tinggi, dan kadang-kadang yang bermodal kuat. Dengan demikian diperlukan suatu tindakan yang berguna untuk menjaga keselamatan data tersebut.

Semakin tinggi nilai data tersebut, semakin tinggi pula tingkat proteksi yang harus diberikan. Ada banyak cara untuk melakukan proteksi data, salah satunya menggunakan suatu cara yang disebut dengan proses **Enkripsi.**

Enkripsi adalah sutau proses untuk melakukan proteksi data dengan cara melakukan perubahan isi suatu data sehingga

data tersebut sukar atau tidak bisa dimengerti oleh pihak lain.[2] Perubahan isi data dilakukan melalui suatu aturan-aturan atau teknik-teknik tertentu. Supaya isi data yang telah berubah dapat dibaca kembali, maka harus dilakukan perubahan isi data ke bentuk asalnya, proses ini disebut dengan **Dekripsi.** Dekripsi dilakukan dengan jalan membalik proses perubahan isi data sesuai dengan aturan atau teknik yang digunakan pada saat enkripsi.

Pada umumnya proses enkripsi digunakan dalam sistem komunikasi data (pengiriman dan penerimaan data), baik komunikasi lewat telepon, komputer, satelit, dan sebagainya, dengan tujuan selain untuk pengamanan data, juga untuk memperkecil ukuran data yang dikirim (teknik pemadatan) sehingga dapat mempercepat waktu dan menghemat biaya pengiriman.

Karena begitu pentingnya kebutuhan akan perlindungan data maka telah dikembangan sebuah algoritma enkripsi salah satunya yang disebut dengan nama DES *(Data Encryption Standard).*

A. Kriptografi Tradisional

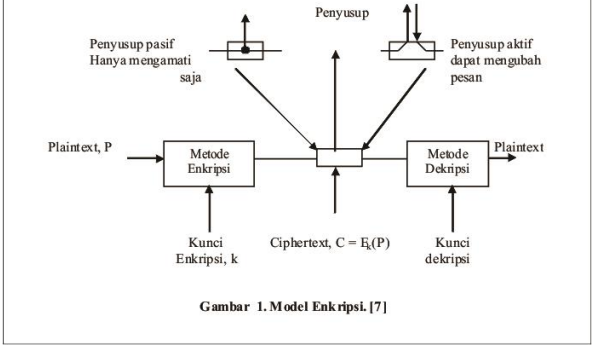
Kriptografi memiliki sejarah yang panjang. Menurut sejarahnya, kriptografi telah digunakan dan disumbang pemikirannya oleh empat kelompok orang : militer, korps, diplomatik, diarist, dan orang yang sedang jatuh cinta. Dari keempatnya, militer telah memainkan perannya yang paling penting dan telah mengembangkan bidang ini. Di dalam organisasi militer, pesan-pesan yang telah di-encode secara tradisional diberikan kepada pekerja code yang berupah buruk untuk selanjutnya dienkripsi dan ditransmisikan. Tugas ini diusahakan agar tidak dilakukan oleh spesialis yang elite. Kendala tambahan telah menjadi kesulitan dalam peralihan yang cepat dari satu metode kriptografi ke metode lainnya, karena hal ini memerlukan pelatihan orang dalam jumlah banyak. Keadaan yang bertolak-belakang ini telah membentuk model enkripsi seperti pada Gambar :

Pesan yang harus dienkripsi, yang dikenal sebagai **plaintext**, ditransformasikan oleh suatu fungsi yang diparameterisasi oleh suatu **kunci**. Kemudian output proses enkripsi, yang dikenal sebagai **ciphertext**, ditransmisikan. Dengan asumsi bahwa musuh, atau **penyusup**, akan mendengar dan secara akurat akan menyalin ciphertext dengan lengkap. Tidak seperti halnya penerima yang dituju, ia tidak mengetahui kunci dekripsi, karena itu tidaklah mudah untuk mendekripsi ciphertext. Kadang-kadang penyusup tidak hanya dapat mendengarkan saluran komunikasi saja (penyusup pasif), namun juga dapat merekam pesan dan mendengarkannya, memasukkan pesan, atau memodifikasi pesan-pesan yang sah sebelum sampai ke penerima (penyusup aktif). Seni memecahkan sandi disebut **Criptanalysis**. Seni memasang sandi (**Criptography**) dan memecahkannya (**Criptanalysis**) secara kolektif disebut **Cryptology.**

Terdapat sejumlah notasi yang berkaitan dengan plaintext, ciphertext, dan kunci. Langkah awal adalah menggunakan $C = E_k(P)$ untuk mengartikan bahwa enkripsi plaintext P yang menggunakan kunci K akan memberikan ciphertext C. Demikian pula halnya, $P = D_k(C)$ merepresentasikan dekripsi C untuk mendapatkan kembali plaintext. Kemudian fungsi tersebut menjadi :

$$D_k(E_k(P)) = P$$

Notasi ini menyatakan bahwa E dan D merupakan fungsi matematik. Satu-satunya hal yang komplek adalah bahwa keduanya merupakan fungsi yang memiliki dua parameter. Untuk membedakannya dari pesan, telah dituliskan salah satu parameter (kunci) sebagai subskrip, bukan sebagai sebuah argumen.



Gambar 1. Model Enkripsi. [7]

Aturan fundamental kriptografi yaitu seseorang harus mengasumsikan bahwa seorang cryptanalyst menguasai metode umum enkripsi yang digunakan. Dengan kata lain, cryptanalyst mengetahu cara kerja metode enkripsi, E, pada Gambar 1. Jumlah usaha yang diperlukan untuk menemukan, menguji, dan memasang metode baru yang

selalu berkompromi atau berpikir untuk berkompromi dengan metode lama akan menyebabkan metode baru itu menjadi tidak berguna untuk menjaga kerahasiaan.

Kunci terdiri dari string yang relatif pendek yang memilih salah satu dari sejumlah banyak enkripsi penting. Sebaliknya dengan metode umum yang jarang berubah, kunci dapat diubah sesering mungkin. Jadi model dasarnya adalah stabil dan merupakan metode umum yang sudah dikenal luas yang diparameterisasi oleh suatu kunci rahasia yang mudah diubah.

Algoritma tidak dapat diandalkan kerahasiaannya. Dengan menggunakan algoritma yang dipublikasikan, cryptographer bebas melakukan konsultasi dengan sejumlah cryptologis akademis yang berkeinginan untuk menembus sistem sehingga cryptographer dapat mempublikasikan tulisan yang menunjukkan bagaimana cerdiknya mereka. Bila setelah algoritma itu dipublikasikan selama 5 tahun dan tidak ada seorang ahlipun berhasil memecahkannya, maka mungkin algoritma itu cukup solid.

Kerahasiaan sebenarnya terletak pada kunci, dan panjang kunci itu merupakan masalah penting dalam rancangan. Ambil suatu kunci kombinasi yang sederhana. Prinsip umumnya adalah bagaimana memasukkan digit secara berurutan. Setiap orang mengetahui hal ini, namun kunci merupakan rahasia. Dengan panjang kunci yang dua digit berarti bahwa terdapat 100 kemungkinan. Panjang kunci tiga digit mempunyai 1000 kemungkinan, dan panjang kunci enam digit mempunyai sejuta kemungkinan. Semakin panjang kunci, semakin tinggi faktor kerja yang harus dilakukan cryptanalyst. Faktor kerja untuk menembus sistem dengan pencarian kunci yang melelahkan merupakan eksponensial terhadap panjang kuncinya. Kerahasiaan berasal dari adanya algoritma yang kuat (namun dipublikasikan) dan kunci yang panjang.

B. Tujuan

Penelitian ini bertujuan untuk membuat sebuah aplikasi yang dapat digunakan untuk proteksi data dengan cara enkripsi dengan menggunakan algoritma DES.

C. Data Encription Standard (DES) / Standard Data Rahasia/Standard Penyandian Data

Penggunaan data sandi yang paling banyak didasarkan pada standard data sandi (DES) yang diambil pada tahun 1977 oleh standard-standard Nasional Bureau, yang sekarang Institut Nasional Standard-Standard dan Tekhnologi (NIST), sebagai Standard Proses Informasi Umum. Untuk DES, data disandikan ke dalam 64 balok bit menggunakan 56 bit kunci. Transformasi-transformasi algoritma 64 bit input ke dalam satu seri langkah-langkah ke dalam 64 bit output. Langkah yang sama dengan kunci yang sama, digunakan untuk cadangan persandian.

DES telah digunakan secara luas dengan peningkatan secara menggembirakan. Sayangnya, ini masih menjadi bahan yang banyak menimbulkan pertentangan, bagaimana memastikan DES itu sendiri. Untuk Lebih jelasnya berikut dijelaskan sedikit sejarah DES.

Di akhir tahun 60-an IBM membangun suatu proyek penelitian dalam Cryptografi komputer, dipimpin oleh Horst Feistel. Proyek di selesaikan tahun 1971 dengan perkembangan sebuah algoritma yang menghasilkan desain LUCIFER yang dijual kepada Loyd's Lonelon, untuk digunakan dalam sistem bagi hasil, juga dikembangkan oleh IBM. LUCIFER adalah suatu blok yang paling murah yang dioperasikan pada blok-blok 64 bit, menggunakan ukuran kunci 128 bit. karena hasil-hasilnya yang menjanjikan yang diproduksi oleh proyek LUCIFER, IBM meluncurkan suatu usaha untuk mengembangkan suatu pasar komersial produk persandian yang idealnya dapat diterapkan pada satu chip.

Usaha itu dipimpin oleh Walter Tuchman dan Carl Meyer dan itupun tidak termasuk hanya riset-riset IBM tapi juga konsultan-konsultan luar dan juga nasihat-nasihat tekhnis dari NSA.

Hasil dari usaha ini disaring dalam versi LUCIFER yang lebih banyak penghambatnya untuk menganalisis persandian, tetapi itu mengurangi ukuran kunci 56 bit. Dengan begitu sesuai dimasukan ke dalam satu chip. Bersamaan itu Standard

National Bureau (NBS) pada tahun 1973 mengumumkan suatu permintaan proposal untuk suatu standard nasional yang paling murah. IBM dipertemukan pada hasil proyek yang dilakukan oleh Tuchman Meyer. Ini jauh lebih baik dari algoritma yang diajukan dan telah diambil sebagai standard data enkripsi pada tahun 1977.

Sebelum DES diambil sebagai standard, pengusulan DES dapat dikritik yang sampai hari ini tidak pernah reda. Dua area digambarkan sebagai kritik yang bertentangan. Pertama, panjang kunci pada algoritma LUCIFER yang baru dari IBM adalah 128 bit, tetapi sistem yang diusulkan hanya 56 bit, suatu pengurangan yang luar biasa dalam ukuran kunci 72 bit. Para pengritik takut (masih takut) untuk berpihak kepada hal di atas bahwa panjang kunci ini terlalu pendek. Area kedua tentang hal ini yang mana kriteria desain untuk struktur internal DES, S-boxes, sedang dan masih diklasifikasikan. Oleh karena itu, para pengguna tidak bisa yakin bahwa struktur internal DES bebas dari titik-titik kelemahan yang tersembunyi yang akan memungkinkan NSA untuk memecahkan pesan-pesan tanpa mengambil faedah dari kunci itu. Peristiwa-peristiwa berikut, khususnya kerja yang terakhir pada *Cryptanalisis deferensial*, rasanya mengindikasikan bahwa DES mempunyai struktur internal yang sangat kuat.

Lebih jauh, berdasarkan orang-orang yang ikut ambil bagian di IBM, satu-satunya perubahan yang dibuat adalah usulan pergantian S-boxes, disarankan oleh NSA, bahwa proses evaluasi itu lama kelamaan diidentifikasikan hal yang sifatnya cepat berubah.

Apapun pangkal pokok masalah itu, DES telah berkembang di tahun-tahun terakhir dan digunakan secara luas, khususnya di dalam aplikasi keuangan. Pada tahun 1994, NIST " membenarkan sekali lagi" DES untuk menggabungkan kegunaan dalam waktu 5 tahun yang lain, NIST merekomendasikan penggunaan DES untuk aplikasi yang lain dari pada perlindungan klasifikasi informasi. Pencipta merasa bahwa, kecuali dalam wilayah-wilayah ekstrem yang sensitif, penggunaan DES dalam aplikasi komersial tidak harus menjadi satu sebab untuk cemas tentang tanggungjawab para manajer.

D. Kelebihan DES

1. DES oleh pemerintah Amerika Serikat telah ditetapkan sebagai cipher produk yang dibuat oleh IBM sebagai standard resmi informasi yang tidak rahasia.
2. Algoritma DES mempunyai 19 tahap yang berlainan, dengan 16 tahap berfungsi identik namun diparameterisasi oleh fungsi-fungsi kunci lainnya.
3. Pada setiap iterasi yang berjumlah 16 itu digunakan kunci yang berbeda, sehingga sulit untuk ditembus.

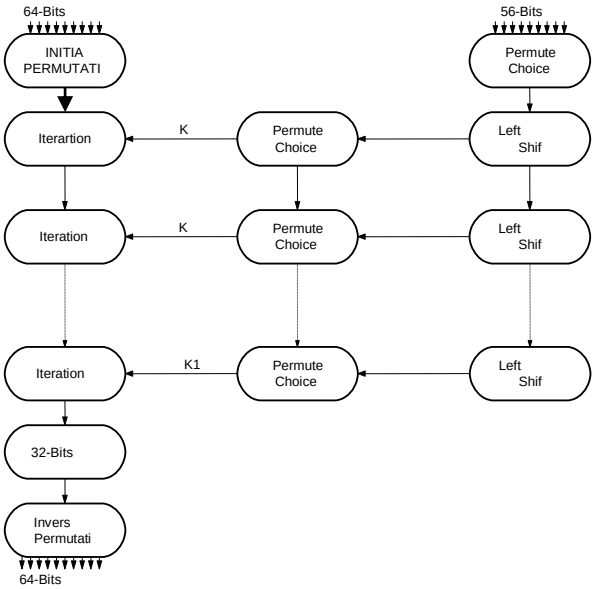
E. Kelemahan DES

Pada dasarnya DES merupakan cipher substitusi monoalfabetis yang menggunakan blok ciphertext 64-bit yang sama. Jadi bagi seorang cryptanalyst yang mengetahui sifat cipher substitusi monoalfabetis dapat digunakan untuk menembus DES.

II. METODOLOGI

2.1 Metode Enkripsi

Metode yang digunakan dalam membuat aplikasi ini dengan metode algoritma DES (*Data Encription Standart*). Algoritma DES mempunyai tahapan proses sebagai berikut:



Gambar 2. Tahapan Kinerja Algoritma DES [2]

Sebagaimana dengan bagan enkripsi, ada dua input untuk fungsi enkripsi, yaitu **Plaintext**, **Kunci** dienkripsikan, pada masalah

ini plaintext panjangnya harus menjadi 64 bit dan kunci panjangnya 56 bit. Melihat sisi kiri dari bagan dapat dilihat bahwa proses *plaintext* berlangsung ke dalam tiga tahap. Pertama, 64 bit *plaintext* melewati satu permutasi awal (IP) yang disusun kembali bitnya untuk menghasilkan *permuted input*. Ini diikuti oleh satu tahap yang terdiri 16 iterasi dari fungsi yang sama yang menyertakan kedua fungsi permutasi dan substitusi. *Output* dari iterasi terakhir (ke 16) terdiri 64 bit yang itu semua adalah satu fungsi dan *input plaintext*. Kiri dan kanan membagi dua *output* yang ditukar untuk menghasilkan *preoutput*. Akhirnya, *preoutput* dilewati satu permutasi (IP-1) sebaliknya satu fungsi inisial permutasi untuk menghasilkan 64 bit *ciphertext*.

Bagian kanan dari gambar 1. menunjukkan cara, yaitu 56 bit kunci digunakan. Pertama-tama kunci dilewati satu fungsi permutasi. Kemudian untuk setiap 16 iterasi, sebuah *subkey* (Ki) dihasilkan oleh kombinasi satu geser putar kiri dan satu permutasi. Fungsi permutasi itu sama untuk setiap iterasi, tetapi satu *subkey* yang berbeda dihasilkan sebab pengulangan pertukaran kunci bit.

2.2 Permutasi Awal

Permutasi awal dan kebalikannya didefinisikan dalam tabel, seperti yang ditunjukkan dalam tabel 2. (a) dan (b), berturut-turut. Untuk mengetahui dua fungsi permutasi ini adalah sungguh-sungguh kebalikan satu sama lain, pertimbangkan 64 bit *input* M berikut ini :

M_1 M_2 M_3 M_4 M_5 M_6 M_7 M_8 M_9 M_{10} M_{11} M_{12} M_{13} M_{14} M_{15} M_{16}
M_{17} M_{18} M_{19} M_{20} M_{21} M_{22} M_{23} M_{24} M_{25} M_{26} M_{27} M_{28} M_{29} M_{30} M_{31} M_{32}
M_{33} M_{34} M_{35} M_{36} M_{37} M_{38} M_{39} M_{40} M_{41} M_{42} M_{43} M_{44} M_{45} M_{46} M_{47} M_{48}
M_{49} M_{50} M_{51} M_{52} M_{53} M_{54} M_{55} M_{56} M_{57} M_{58} M_{59} M_{60} M_{61} M_{62} M_{63} M_{64}

dimana Mi adalah satu binary digit. Kemudian permutasi X = IP(M) adalah sebagai berikut :

M_{58} M_{50} M_{42} M_{34} M_{26} M_{18} M_{10} M_2 M_{60} M_{52} M_{44} M_{36} M_{28} M_{20} M_{12} M_4
M_{62} M_{54} M_{46} M_{38} M_{30} M_{22} M_{14} M_6 M_{64} M_{56} M_{48} M_{40} M_{32} M_{24} M_{16} M_8
M_{57} M_{49} M_{41} M_{33} M_{25} M_{17} M_9 M_1 M_{59} M_{51} M_{43} M_{35} M_{27} M_{19} M_{11} M_3
M_{61} M_{53} M_{45} M_{37} M_{29} M_{21} M_{13} M_5 M_{63} M_{55} M_{47} M_{39} M_{31} M_{23} M_{15} M_7

Jika kemudian mengambil kebalikan permutasi $Y = IP^{-1}(X) = IP^{-1}(IP(M))$, ini dapat dilihat bahwa pemesanan-pemesanan bit semula dapat dikembalikan.

TABEL 2 Tabel Permutasi DES

(a) Permutasi Awal (IP)

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

(b) Balikan Permutasi Awal (IP^{-1})

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

(c) Permutasi Ekspansi (E)

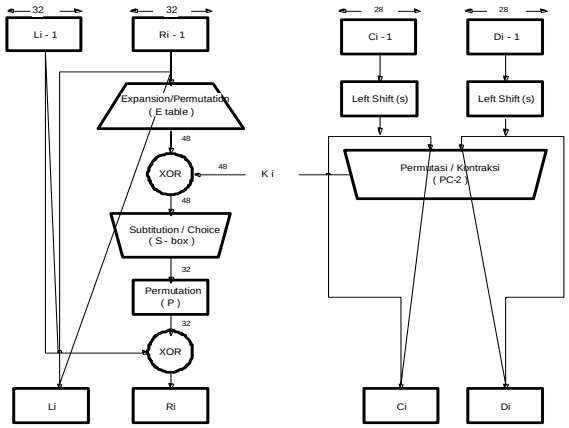
Keluaran bit	1	2	3	4	5	6	7	8	9	10	11	12
Dari masukan	32	1	2	3	4	5	4	5	6	7	8	9
Keluaran bit	13	14	15	16	17	18	19	20	21	22	23	24
Dari masukan	8	9	10	11	12	13	12	13	14	15	16	17
Keluaran bit	25	26	27	28	29	30	31	32	33	34	35	36
Dari masukan	16	17	18	19	20	21	20	21	22	23	24	25
Keluaran bit	37	38	39	40	41	42	43	44	45	46	47	48
Dari masukan	24	25	26	27	28	29	28	29	30	31	32	1

(d) Fungsi Permutasi (P)

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

2.3 Rincian Iterasi Tunggal

Sekarang dapat dilihat lebih dekat lagi pada algoritma iterasi tunggal atau satu iterasi sebagaimana diilustrasikan pada gambar 3. berikut :



Gambar 3. Rincian Iterasi Tunggal [2]

Dimulai dengan memfokuskan sisi kanan dari diagram. Pada intinya, 64 bit *input* dipermutasikan melalui 16 iterasi, Pada kesimpulannya disetiap iterasi menghasilkan satu nilai menengah 64 bit. Separo kanan dan kiri dari setiap nilai menengah 64 bit diperlakukan secara terpisah sejumlah 32 bit, diberi label L (kiri) dan R (kanan).[3] Proses keseluruhan pada setiap iterasi dapat dijumlahkan ke dalam rumus berikut ini :

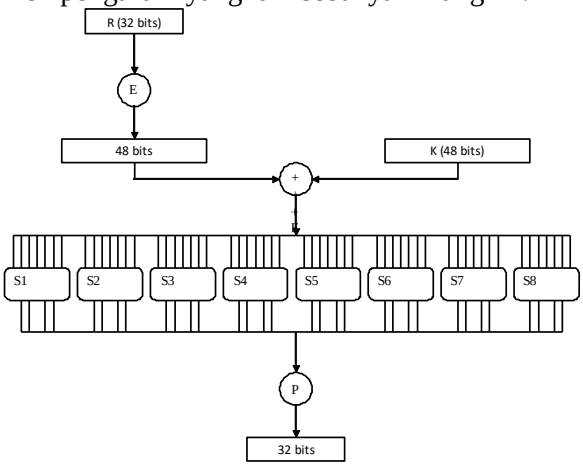
$$L_i = R_{i-1}$$

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

Dimana $\oplus$ menunjukkan fungsi lebar bit XOR. Jadi output bagian kiri dari satu iterasi (L_i) secara sederhana sama dengan input bagian kanan iterasi (R_{i-1}). Output sebelah kanan (R_i) adalah exclusive atau *(XOR)* dari L_{i-1} dan satu fungsi komplek f atau R_{i-1} dan K_i. Fungsi f adalah digambarkan dalam gambar 3. Kunci iterasi K_i adalah 48 bit. Input R adalah 32 bit input, ini adalah pertama diperluas menjadi 48 bit mencari satu tabel yang didefinisikan satu perluasan tambahan perluasan permutasi yang meliputi turunan dari nilai 16 bit dari bit R. Menghasilkan 48 bit adalah di XOR dengan K_i. Hasil 48 bit ini melalui satu fungsi pengganti yang menghasilkan 32 bit output yang dipermutasikan. Substitusi itu terdiri dari satu set dari 8 S-boxes, pada setiap dari yang menerima 6 bit sebagai input dan menghasilkan 4 bit output. Transformasi bit yang pertama dan terakhir dari output ke dalam box S_1, membentuk satu nomor biner 2 bit untuk (memilih/menyeleksi) satu baris tertentu di dalam Tabel, untuk S_1. Pertengahan 4 bit (memilih/menyeleksi) satu kolom tertentu. Nilai desimal dalam sel dipilih oleh baris dan kolom kemudian dirubah ke dalam 4-bitnya untuk menghasilkan output.

Gambar 5. menyediakan operasi *S-boxes* secara detail yang mungkin berguna dalam memahami pemetaan. Bit-bit yang pertama dan terakhir dari input dan memilih satu dari empat permutasi yang ditegaskan oleh baris-baris dalam Tabel *S-boxes*. Gambar itu menunjukkan permutasi untuk baris 0 dari Box S_1

Susunan dari *S-Boxes–S-boxes* adalah ulasan lebih lanjut yang cukup baik. Abaikan untuk sementara sumbangan dari kunci atau K_i, sekarang jika menguji tabel pengembangan maka akan tahu bahwa 32 bit input dibagi ke dalam kelompok yang tediri dari 4 bit dan kemudian menjadi kelompok yang terdiri dari 6 bit dengan mengambil keluar bit-bit dari dua kelompok yang berdekatan. Keluaran dua bit dari setiap kelompok memilih satu dari empat kemungkinan tabel substitusi. Kemudian satu nilai 4 bit output disubstitusikan untuk input 4 bit tertentu (pertengahan 4 input bit). 32 bit output dari 8 *S-Boxes* kemudian dipermutasikan dalam iterasi output yang akan terjadi dari setiap *S-Boxes* dengan segera mempengaruhi yang lain sebanyak mungkin.

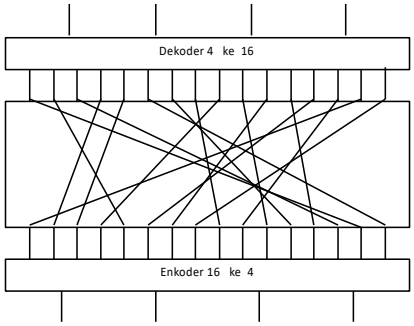


Gambar 4. Subtitusi (S-box) Perhitungan R dan K [4]

TABEL 3. Definisi Kotak-S DES

Nomor Kolom

Baris	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Kotak
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	S1
2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0	
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13	
0	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10	
1	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5	S2
2	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15	
3	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9	
0	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8	
1	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1	S3
2	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7	
3	1	10	13	0	6	9	8	7	4	15	16	3	11	5	2	12	
0	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15	
1	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9	S4
2	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4	
3	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14	
0	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9	
1	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6	S5
2	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14	
3	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3	
0	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11	
1	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8	S6
2	9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6	
3	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13	
0	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1	
1	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6	S7
2	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2	
3	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12	
0	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7	
1	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2	S8
2	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8	
3	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11	



Gambar 5. Rincian Kotak S (S-Box)[2]

2.4. Pembangkitan Kunci

Kembali ke bagan /gambar 2. yaitu bahwa kunci 56 bit digunakan sebagai *input* ke dalam algoritma, adalah persoalan pertama untuk suatu permutasi *Choice One* (permutasi pilihan pertama) (Tabel 4.(a).). Hasil kunci 56 bit adalah kemudian diperlakukan sebagai dua jumlah 28 bit, dinamakan C_o dan D_o. Di setiap iterasi, C dan D adalah dijalankan secara terpisah ke dalam satu penggeseran ke kiri bundaran atau rotasi dari 1 atau 2 bit sebagaimana di tentukan oleh (Tabel 4.(c).). Nilai penggeseran ini menyajikan sebagai input ke dalam iterasi berikutnya.

Itu juga menyajikan sebagai *input* untuk *Permuted Choice Two* (Permutasi Pilihan ke Dua). Tabel 4 (b).), yang menghasilkan satu 48 bit output yang menyajikan sebagai input ke dalam fungsi $f(R_{i-1}, K_i)$.

TABEL 4. Tabel Untuk Perhitungan Jadwal Kunci DES

(a) Permutasi Pilihan 1(PC-1)

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

(b) Permutasi Pilihan 2 (PC-2)

Keluaran bit	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Dari masukan	14	17	11	24	1	5	3	28	15	6	21	10	23	19	12	4
Keluaran bit	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
Dari masukan	26	8	16	7	27	20	13	2	41	52	31	37	47	55	30	40
Keluaran bit	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
Dari masukan	51	45	33	48	44	49	39	56	34	53	46	42	50	36	29	32

(c) Penjadwalan Geser Kiri

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

2.5. Metode Dekripsi (DES)

Proses dekripsi dengan DES pada dasarnya sama dengan proses enkripsi. Aturannya adalah sebagai berikut : Gunakan *ciphertext* sebagai *input* ke dalam Algoritma DES, namun gunakan kunci-kunci K_i dalam urutan yang terbalik. Itu adalah gunakan K_{16} pada iterasi pertama, K_{15} pada iterasi ke dua, dan seterusnya sampai K_1 digunakan pada iterasi Ke-16 dan iterasi terakhir.

Untuk melihat algoritma yang sama dengan kunci yang urutannya terbalik menghasilkan hasil yang benar, dapat dilihat gambar 2 yang menunjukkan proses enkripsi berlangsung ke bawah sisi kiri dan proses dekripsi berlangsung ke atas sisi kanan. Diagram menunjukkan bahwa, pada setiap tingkat nilai menengah dari proses dekripsi adalah sama mulai hubungannya dari proses enkripsi dengan membagi dua dari nilai tukar, untuk menyimpan yang lain ini, dari *output* tingkat enkripsi ke-i dijadikan Li || Ri (Li dirangkaikan dengan Ri). Kemudian menghubungkan input ke dalam tingkatan (16

- i) tingkat dekripsi adalah Li || Ri.

Setelah iterasi terakhir dari proses enkripsi, dua bagian dari output adalah ditukar, sehingga input tersebut dalam tingkat IP^{-1} terakhir adalah $R_{16} || L_{16}$. *Output* dari tingkat tersebut adalah *ciphertext*. Sekarang pakai *ciphertext* dan gunakan *ciphertext* tersebut sebagai *input* ke dalam algoritma DES. Tahap pertama adalah melewati *ciphertext* melalui tingkat IP, menghasilkan jumlah 64-bit $L^d_0 || R^d_d$. Tetapi mestinya harus sudah tahu bahwa IP adalah kebalikan dari IP^{-1} ;

di mana ;

$$L^d_0 || R^d_d = IP(\text{ciphertext})$$

$$\text{Ciphertext} = IP^{-1}(R_{16} || L_{16})$$

$$L^d_0 || R^d_d = IP(IP^{-1}(R_{16} || L_{16})) = R_{16} || L_{16}$$

Sehingga input dari tingkatan pertama dari dekripsi proses adalah sama ke dalam 32-bit tukar menukar dari output tingkatan Ke-16 dari proses enkripsi.

Sekarang menunjukkan bahwa output tingkat pertama dari proses dekripsi adalah sama dengan tukar menukar 32-bit dari tingkat ke-16 input dari proses enkripsi pertama, pertimbangkan proses enkripsi, bahwa :

$$L_{16} = R_{15}$$

$$R_{16} = L_{15} \oplus f(R_{15}, K_{16})$$

Pada sisi dekripsi:

$$L^d_1 = R^d_0 = L_{16} = R_{16}$$

$$R^d_1 = L^d_0 \oplus f(R^d_0, K_{16})$$

$$= R_{16} \oplus f(R_{15}, K_{16})$$

$$= [\ L_{15} \oplus f(R_{15}, K_{16})\ ] \oplus f(R_{15}, K_{16})$$

XOR mengikuti sifat :

$$[\ A \oplus B\ ] \oplus C = A \oplus [\ B \oplus C\ ]$$

$$D \oplus D = 0$$

$$E \oplus 0 = E$$

Sehingga akan mempunyai $L^d_1 = R_{15}$ dan $R^d_1 = L_{15}$. Oleh karena itu output tingkat pertama dari proses dekripsi adalah $L_{16} || R_{16}$ dimana pertukaran 32 bit input ke dalam tingkatan ke-16 dari enkripsi. Persesuaian ini menyimpan semua jalan melewati iterasi ke-16, sebagaimana dengan mudah ditunjukkan, dapatlah didudukkan proses ini ke dalam peryataan umum. Untuk tingkat iterasi ke-i dari enkripsi Algoritma :

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

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

Menyusun kembali pernyataan-pernyataan:

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

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

Sehingga, telah dilukiskan input-input ke dalam iterasi ke-i sebagai satu fungsi dari output-output, persamaan ini memperkuat penempatan-penempatan yang ditunjukkan dalam sisi kanan dari gambar 2.

Akhirya, tahu bahwa output dari tingkat terakhir proses dekripsi adalah $R_0 || L_d$. Satu pertukaran 32 bit ditampilkan, sehingga input dalam tingkat IP^{-1} adalah $L_0 || R_d$. Tetapi, $IP^{-1}(L_0 || R_d) = IP^{-1}$ (IP(plaintext)) = Plaintext.

sehingga plaintext yang asli diperoleh, mendemonstrasikan ketepatan proses dekripsi DES.

III. HASIL dan PEMBAHASAN

Program Enkripsi dan Dekripsi dengan algoritma DES ini bisa digunakan untuk mengenkripsi atau mendekripsi semua file kecuali file video. Berikut akan diuji coba beberapa file yang mempunyai tipe berlainan dan juga kapasitasnya, dalam satuan byte dan perbandingan per-1024 byte kelipatan. Program ini juga dilengkapi dengan waktu akses pada saat proses Enkripsi ataupun Dekripsi (dalam satuan detik), dan juga Grafik kecepatan akses per-kapasitas file. Grafik kecepatan akses enkripsi/dekripsi penulis memberikan batasan kapasitas file maksimum 5 Kbyte atau 5120 byte. Berikut disajikan tabel 1 nama file yang telah diuji, beserta grafik kecepatan proses enkripsi ataupun dekripsi. Grafik laju kecepatan proses enkripsi atau dekripsi, mempunyai tingkat pertambahan waktu yang sama antara file satu dengan file yang lainya, yaitu kelipatan 3 detik. Untuk itu penulis hanya menyajikan 1 gambar grafik perbandingan proses enkripsi dan dekripsi.

Tabel 5. Uji coba file proses enkripsi dan dekripsi

NAMA FILE	EKSTENSI	JENIS FILE	KETERANGAN

ISSN : 1858 - 4144

Latih1	TXT	Teks	Dibuat dengan editor DOS
Latih2	TXT	Teks	Dibuat dengan editor Windows (Notepad)

Contoh Plaintext dan hasil Enkripsi/Dekripsi-nya
Nama file : LATIH1.TXT

```
A:\>TYPE LATIH1.TXT
ini percobaan program untuk demo
semoga berhasil dengan sukses
123345 adalah angka percobaan
terimakasih
sukses selalu amin!!!!!!
File ini untuk percobaan...
Pada 1977, dua orang peneliti kriptografi Universitas Stanford, Diffie dan Hellm
an (1977), merancang
mesin penembus DES dan mengestimasi biaya pembuatannya sebesar 20 juta dollar. B
ila potongan kecil
plaintext telah ditemukan dan cocok dengan ciphertext, maka mesin ini akan dapat
 menemukan dengan
menggunakan pencarian secara menyeluruh 2^56 entry kunci dalam waktu kurang dari
 1 hari. Saat ini,
mesin seperti itu mungkin hanya memerlukan biaya 1 juta dollar. Rancangan detail
 mesin yang dapat menembus
DES dengan pencarian secara menyeluruh dalam waktu sekitar empat jam dijelaskan
dakam (Wiener,1994)
Strategi lainnya untuk menembus DES. Walaupun enkripsi Sofware 1000 kali lebih l
ambat dibanding dengan enkripsi
hardware, komputer rumah yang baik masih dapat melakukan enkripsi 250.000 enkrip
si/detik.
```

Dienkripsikan pada file LATIH1.ENC, dengan kunci STMIKYK

```
PROGRAM ENKRIPSI / DEKRIPSI

JENIS [E]NKRIPSI / [D]EKRIPSI  : D
FILE YANG AKAN DI-DEKRIPSI     : LATIH1.ENC
DI-DEKRIPSI -KAN KE FILE       : LATIH1.DKP
DENGAN KUNCI                   : STMIKYK
```

Didekripsikan pada file LATIH1.DKP

```
PROGRAM ENKRIPSI / DEKRIPSI

JENIS [E]NKRIPSI / [D]EKRIPSI  : E
FILE YANG AKAN DI-ENKRIPSI     : LATIH1.TXT
DI-ENKRIPSI -KAN KE FILE       : LATIH1.ENC
DENGAN KUNCI                   : STMIKYK

A:\>TYPE LATIH1.ENC
êÿ╟▓Åq╞|±≡÷eöXIEusé£╪|R°
```

```
A:\>TYPE LATIH1.DKP
ini percobaan program untuk demo
semoga berhasil dengan sukses
123345 adalah angka percobaan
terimakasih
sukses selalu amin!!!!!!
File ini untuk percobaan...
Pada 1977, dua orang peneliti kriptografi Universitas Stanford, Diffie dan Hellm
an (1977), merancang
mesin penembus DES dan mengestimasi biaya pembuatannya sebesar 20 juta dollar. B
ila potongan kecil
plaintext telah ditemukan dan cocok dengan ciphertext, maka mesin ini akan dapat
 menemukan dengan
menggunakan pencarian secara menyeluruh 2^56 entry kunci dalam waktu kurang dari
 1 hari. Saat ini,
mesin seperti itu mungkin hanya memerlukan biaya 1 juta dollar. Rancangan detail
 mesin yang dapat menembus
DES dengan pencarian secara menyeluruh dalam waktu sekitar empat jam dijelaskan
dakam (Wiener,1994)
Strategi lainnya untuk menembus DES. Walaupun enkripsi Sofware 1000 kali lebih l
ambat dibanding dengan enkripsi
hardware, komputer rumah yang baik masih dapat melakukan enkripsi 250.000 enkrip
si/detik.
```

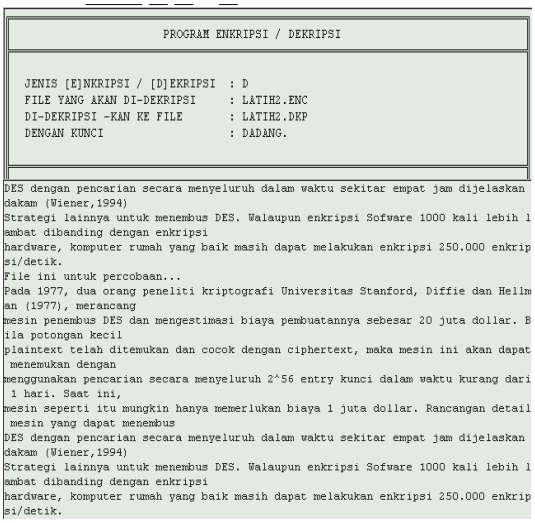
Nama File : LATIH2.TXT

Dienkripsikan pada file LATIH2.ENC, dengan Kunci DADANG.

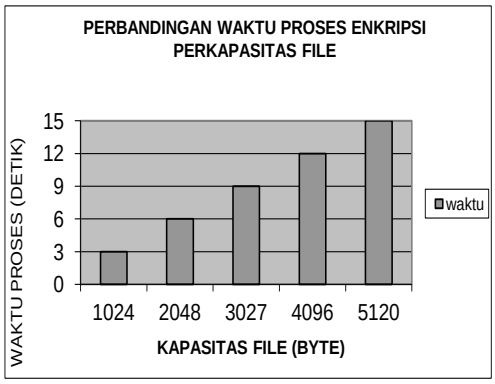
Nama File : LATIH2.TXT

```
PROGRAM ENKRIPSI / DEKRIPSI

JENIS [E]NKRIPSI / [D]EKRIPSI  : E
FILE YANG AKAN DI-ENKRIPSI     : LATIH2.TXT
DI-ENKRIPSI -KAN KE FILE       : LATIH2.ENC
DENGAN KUNCI                   : DADANG.

A:\>TYPE LATIH2.ENC
↔ó!}1v+b╞╜w8>─[│Æì╜ö╞8┘┘}┐║╣σ∞~┴y┤2¡ì╖÷bViHε ┌Æ=ënßµëß╖¬è─J _‡!)P▼obQτÅΓéδ╩╖dïf
α╩⌠ ▀¶·╧·1╟y▌ⁿG║Hℓφ‡)╪♦N▲s!SüÄ<♣│Σ║α╜ ÜxÖℓ"Ωfàℓ ½

A:\>TYPE LATIH2.TXT_
DES dengan pencarian secara menyeluruh dalam waktu sekitar empat jam dijelaskan
dakam (Wiener,1994)
Strategi lainnya untuk menembus DES. Walaupun enkripsi Sofware 1000 kali lebih l
ambat dibanding dengan enkripsi
hardware, komputer rumah yang baik masih dapat melakukan enkripsi 250.000 enkrip
si/detik.
File ini untuk percobaan...
Pada 1977, dua orang peneliti kriptografi Universitas Stanford, Diffie dan Hellm
an (1977), merancang
mesin penembus DES dan mengestimasi biaya pembuatannya sebesar 20 juta dollar. B
ila potongan kecil
plaintext telah ditemukan dan cocok dengan ciphertext, maka mesin ini akan dapat
 menemukan dengan
menggunakan pencarian secara menyeluruh 2^56 entry kunci dalam waktu kurang dari
 1 hari. Saat ini,
mesin seperti itu mungkin hanya memerlukan biaya 1 juta dollar. Rancangan detail
 mesin yang dapat menembus
DES dengan pencarian secara menyeluruh dalam waktu sekitar empat jam dijelaskan
dakam (Wiener,1994)
Strategi lainnya untuk menembus DES. Walaupun enkripsi Sofware 1000 kali lebih l
ambat dibanding dengan enkripsi
hardware, komputer rumah yang baik masih dapat melakukan enkripsi 250.000 enkrip
si/detik.
```

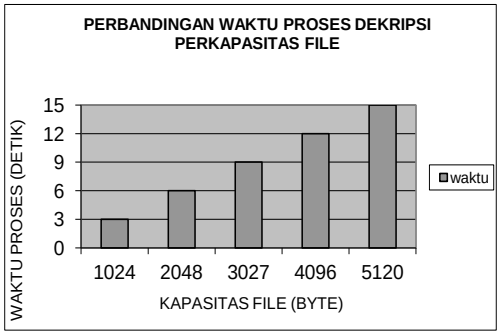
```
PROGRAM ENKRIPSI / DEKRIPSI

JENIS [E]NKRIPSI / [D]EKRIPSI :  E
FILE YANG AKAN DI-ENKRIPSI     : LATIH2.TXT
DI-ENKRIPSI -KAN KE FILE       : LATIH2.ENC
DENGAN KUNCI                   : DADANG.

A:\>TYPE LATIH2.ENC
↔ó!}1v+b╞╜w8>─[│Æì╜ö╞8┘┘}┐║╣σ∞~┴y┤2¡ì╖÷bViHε ┌Æ=ënßµëß╖¬è─J _‡!)P▼obQτÅΓéδ╩╖dïf
α╩⌠ ▀¶·╧·1╟y▌ⁿG║Hℓφ‡)╪♦N▲s!SüÄ<♣│Σ║α╜ ÜxÖℓ"Ωfàℓ ½
A:\>
```



Gambar berikut, menjelaskan perbandingan waktu proses enkripsi dan dekripsi sesuai dengan
kapasitas file.



Gambar 6. Grafik Perbandingan Waktu Proses Enkripsi Perkapasitas File.



Gambar 7. Grafik Perbandingan Waktu Proses Dekripsi Perkapasitas File.

IV. KESIMPULAN DAN SARAN

Dari beberapa pembahasan atau uraian di atas dapatlah diambil kesimpulan bahwa ;

1. Algoritma DES mempunyai 19 tahapan sistem kerja dengan 16 tahapan dikombinasikan dengan kunci, sebanyak 7 karakter sehingga sulit untuk ditembus, dengan demikian DES cukup handal untuk perlindungan keamanan data.
2. Dari hasil pengujian beberapa file menunjukkan bahwa laju kecepatan proses Enkripsi ataupun Dekripsi menunjukkan tingkat pertambahan yang sama (linear), yaitu kelipatan 3 detik per-1024 byte.

Program enkripsi ini menggunakan algoritma DES, dengan satu kali proses enkripsi, sehingga masih dimungkinkan untuk dikembangkan menjadi dua kali atau tiga kali enkripsi atau dengan DES yang dimodifikasi dapat menjadikan program ini lebih sulit untuk ditembus.

DAFTAR PUSTAKA

[1] Tanenbaum, A. S., *Computer Networks*, Prentice Hall, 1997.

[2] Stallings, W., *Network and Internetwork Security Principles and Practice*, Prentice Hall, 1995.

[3] Yuliana, K,D, 2005, Modul Pembelajaran Enkripsi DES Melalui Visualisasi, http://www.gunadarma.ac.id/library/articles/graduate/industrial-technology/2005/Artikel_50401408.pdf

[4] Lai, X., *On the Design and Security of Block Ciphers*, Hartung-Gorre, 1992.

[5] Lai, X., and MASSEY, J., *Advances in Cryptology-Eurocrypt 90 Proceedings*, Springer-Verlag, 1990.