Jurnal Matematika Vol. 16, No. 1, Mei 2017
ISSN: 1412-5056
http://ejournal.unisba.ac.id/

Diterima: 05/01/2017
Disetujui: 30/04/2017
Publikasi: 20/05/2017

Implementasi Algoritma DES Menggunakan MATLAB

Andi Priatmoko, Erwin Harahap

Prodi Matematika, FMIPA, Universitas Islam Bandung,

andipriatmoko.ap@unisba.ac.id, erwin2h@unisba.ac.id

Abstrak. Algoritma *Data Encryption Standard* (DES) adalah algoritma kriptografi yang termasuk dalam algoritma simetrik, dengan menggunakan kunci yang sama untuk enkripsi dan dekripsi. DES menggunakan 16 putaran dengan 16 buah kunci internal yang dapat dibangkitkan dari kunci eksternal yang diberikan oleh pengguna. Kunci eksternal memilik panjang 64 bit digunakan untuk mengenkripsi atau mendekripsi data 64 bit. Proses enkripsi dan dekripsi dari algoritma DES dengan menggunakan software MATLAB mengefisienkan pengguna dalam melakukan proses enkripsi dan dekripsi data. MATLAB dilengkapi dengan fitur *Graphical User Interface* (GUI) yang dapat menyembunyikan kerumitan program, sehingga dengan adanya *interface* algoritma DES, pengguna lebih mudah berinteraksi dengan program.

Kata Kunci: DES, MATLAB, GUI, Chiperteks, Enkripsi, Dekripsi.

Abstract. (*The implementations of DES Algorithms Using MATLAB*) Data Encryption Standard (DES) is a cryptography algorithm that are included in the algorithms symmetric, by using the same for encryption and decryption. DES the use of 16 rounds with 16 pieces of the key that can be raised from the key external given by the user. The key to the external is 64 bits used to encrypt and decrypt data 64 bits. The process of encryption and decryption of algorithms DES by using software MATLAB efficient users in to do the encryption and decryption. MATLAB is equipped with features Graphical User Interface (GUI) that can hide the complexity of the program, so that with the interface of the DES, users are more easily interact with the program.

Keywords: DES, MATLAB, GUI, Chiperteks, Encryption, Decryption.

1. Pendahuluan

Kemudahan akses media komunikasi membawa pengaruh terhadap keamanan informasi yang menggunakan media komunikasi sebagai media penyampaian. Informasi menjadi sangat rentan untuk diketahui, diambil atau bahkan dimanipulasi dan disalahgunakan oleh pihak lain yang tidak berhak. Selama pengiriman dan ketika sampai di tujuan, informasi tersebut harus tetap rahasia dan terjaga keasliannya atau tidak dimodifikasi. Penerima informasi harus yakin bahwa informasi tersebut memang benar berasal dari pengirim yang tepat, begitu juga sebaliknya, pengirim yakin bahwa penerima informasi adalah orang yang sesungguhnya. Selain itu penerima tidak ingin pengirim membantah pernah mengirim informasi tersebut, dan jika hal tersebut terjadi penerima perlu membuktikan ketidakbenaran penyangkalan tersebut. Untuk permasalahan-permasalahan keamanan tersebut diperlukan suatu metode untuk menjaga keamanan informasi. Salah satu metodenya adalah kriptografi.

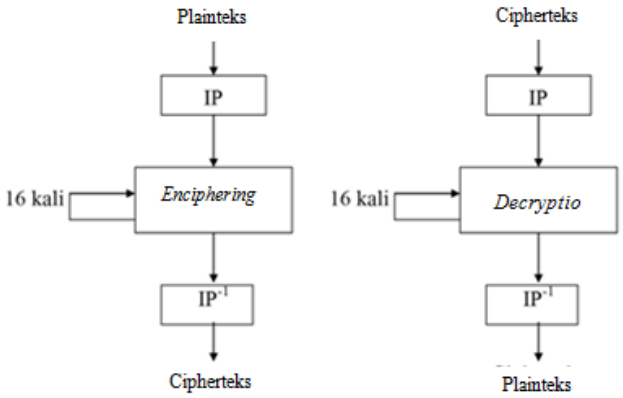
Kriptografi adalah ilmu yang mempelajari teknik-teknik matematika yang berhubungan dengan aspek keamanan data dan informasi seperti keabsahan data, integritas data, serta autentifikasi data. Kriptografi bertujuan untuk mengamankan data atau informasi selama proses pengiriman, dan menjaga agar data atau informasi tersebut sampai pada pihak yang tepat. Salah satu teknik yang dipakai adalah dengan menggunakan kriptografi, yaitu ilmu yang menyandikan suatu pesan menjadi kode tertentu yang sulit dimengerti. Dengan menggunakan kriptografi, pesan asli yang dikirim (*plaintext*) diubah ke dalam bentuk pesan tersandi (*ciphertext*). Kemudian pesan tersandi tersebut dapat dikembalikan ke bentuk pesan sebenarnya hanya dengan menggunakan kunci (*key*) tertentu yang hanya dimiliki oleh pihak yang sah saja.

Saat ini telah banyak bermunculan berbagai algoritma kriptografi yang tentunya setiap algoritma menawarkan kelebihan dan kekurangan masing-masing. Salah satu kriptografi modern algoritma enkripsi kunci simetrik yang paling umum digunakan adalah *Data encryption Standard* (DES). *Data encryption Standard* (DES) adalah algoritma cipher blok yang populer karena dijadikan standard algoritma enkripsi kunci-simetri. Sebenarnya DES adalah namanya standard enkripsi simetri, nama algoritma enkripsinya sendiri adalah DEA (*Data Encryption Algorithm*) dikembalikan di IBM dibawah kepemimpinan W.L. Tuchman pada tahun 1972. Algoritma ini didasarkan pada algoritma *Lucifer* yang dibuat oleh Horst Feistel. Algoritma ini telah disetujui oleh *National Bureau of Standard* (*NBS*) setelah penilaian kekuatannya oleh *National Security Agency* (*NSA*) Amerika Serikat (Rinaldi Munir, 2006). Berdasarkan tujuan dalam penelitian ini di uraikan dalam pokok-pokok yang ingin memahami proses penyandian dengan *Data encryption Standard* (DES), memahami dan mengetahui bagaimana cara merancang sebuah model implementasi algoritma DES untuk enkripsi dan dekripsi data menggunakan *Graphical User Interface* (GUI) MATLAB, dan mengetahui hasil implementasi algoritma DES pada keamanan pesan.

2. Landasan Teori

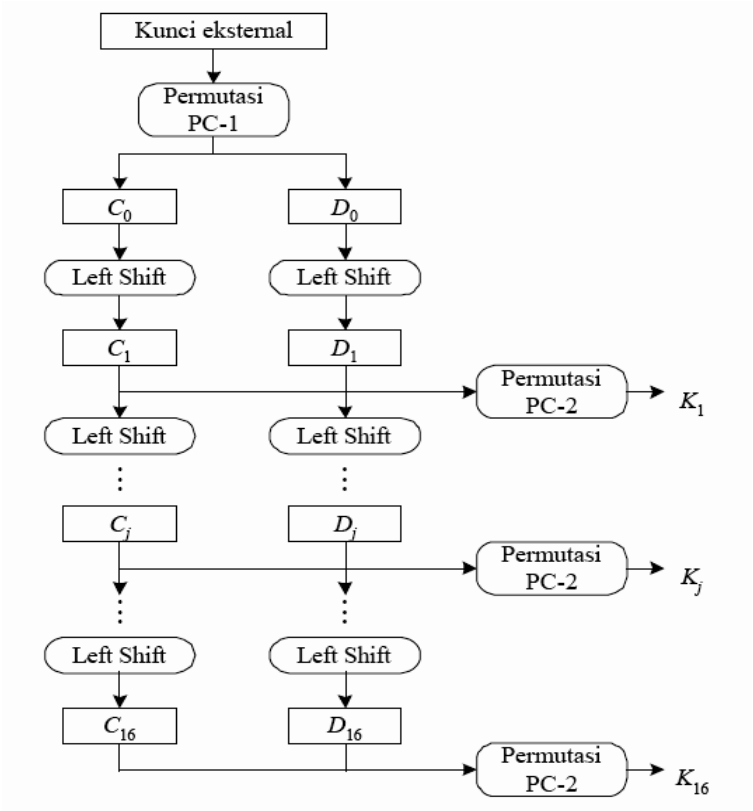
Komponen pada sandi Fiestel adalah memanfaatkan sandi produks yang digunakan secara berulang dalam beberapa ronde. Komponen pada sandi Fiestel dapat bersifat *self-invertible* (invers dengan komponen yang sama), *invertible* (memiliki invers) *non-invertible* (tidak memiliki invers) (Rifki Sadikin, 2012). DES merupakan salah satu contoh sandi Fiestel, sehingga struktur sandi DES memiliki struktur yang sama dengan sandi Fiestel dengan penyususan. Panjang blok DES adalah 64 bit, jadi ukuran teks asli dan teks sandi adalah 64 bit. Ukuran kunci DES adalah 64 bit dan ukuran kunci ronde adalah 48 bit dan jumlah ronde pada DES adalah 16 ronde. Struktur sandi DES dalam skema global adalah sbb:

1. Blok plainteks dipermutasi dengan matriks permutasi awal (*initial permutation* atau IP).
2. Hasil permutasi awal kemudian di-*enciphering*- sebanyak 16 kali (16 putaran). Setiap putaran menggunakan kunci internal yang berbeda.
3. Hasil *enciphering* kemudian dipermutasi dengan matriks permutasi balikan (*invers initial permutation* atau IP^{-1}) menjadi blok cipherteks.



Gambar 1. Skema global DES (Rinaldi Munir, 2006)

Enkripsi dan dekripsi DES memiliki 16 putaran, maka dibutuhkan kunci internal sebanyak 16 buah, yaitu K_1,K_2,…,K_{16}. Kunci-kunci internal ini dapat dibangkitkan sebelum proses enkripsi atau bersamaan dengan proses enkripsi. Kunci internal dibangkitkan dari kunci eksternal yang diberikan oleh pengguna. Kunci eksternal panjangnya 64 bit atau 8 karakter.



Gambar 2. Penurunan kunci DES (Rinaldi Munir, 2006)

Suatu kunci 64 bit digunakan sebagai input kunci akan tetapi tiap bit kedelapan diabaikan akibatnya menghasilkan 56 bit input dan disajikan dalam dua bit string yang berjumlah masing masing 28 bit dan selanjutnya mengalami transformasi left shifts dimana setiap K_i round dilakukan perputaran 1 atau 2 bit sesuai dengan round K_i.

Proses enkripsi terhadap blok plainteks dilakukan setelah permutasi awal (IP). Tujuan permutasi awal adalah mengacak plainteks sehingga urutan bit-bit di dalamnya berubah. Pengacakan dilakukan dengan menggunakan matriks permutasi awal berikut:

Tabel 1. Initial permutation (IP)

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

Setiap blok plainteks mengalami 16 kali putaran enkripsi. Setiap putaran enkripsi merupakan jaringan Feistel yang secara matematis dinyatakan sebagai

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

E adalah fungsi ekspansi yang memperluas blok R_{i-1} yang panjangnya 32-bit menjadi blok 48 bit. Fungsi ekspansi direalisasikan dengan matriks permutasi ekspansi sbb:

Tabel 2. Expansion

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

Selanjutnya, hasil ekpansi, yaitu $E(R_{i-1})$, yang panjangnya 48 bit di-XOR-kan dengan K_i yang panjangnya 48 bit menghasilkan vektor A yang panjangnya 48-bit:

$$E(R_{i-1}) \oplus K_i = A$$

Vektor A dikelompokkan menjadi 8 kelompok, masing-masing 6 bit, dan menjadi masukan bagi proses substitusi. Proses substitusi dilakukan dengan menggunakan delapan buah kotak-S (*S-box*), S_1 sampai S_8. Setiap kotak-S menerima masukan 6 bit dari vektor A dan menghasilkan keluaran 4 bit. Kelompok 6-bit pertama menggunakan S_1, kelompok 6-bit kedua menggunakan S_2, dan seterusnya.

Tabel 3. S-box

S_1:

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

S_2:

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

S_3:

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

S_4:

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

S_5:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
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	16
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

S_6:

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

S_7:

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

S_8:

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

Keluaran proses substitusi adalah vektor *B* yang panjangnya 48 bit. Vektor *B* menjadi masukan untuk proses permutasi. Tujuan permutasi adalah untuk mengacak hasil proses substitusi kotak-S. Permutasi dilakukan dengan menggunakan matriks permutasi *P* (*P-box*) sbb:

Tabel 4. Fungsi permutasi f(P)

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

Selanjutnya, bit-bit *P*(*B*) di-*XOR*-kan dengan L_{i-1} untuk mendapatkan R^i.

$$R_i = L_{i-1} \oplus P(B)$$

Jadi, keluaran dari putaran ke-*i* adalah

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

Permutasi terakhir dilakukan setelah 16 kali putaran terhadap gabungan blok kiri dan blok kanan. Proses permutasi menggunakan matriks permutasi awal balikan (*inverse initial permutation* atau IP^{-1}) sbb:

Tabel 5. Inverse Initial Permutation

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

Sedangkan proses perubahan cipherteks menjadi plainteks disebut proses dekripsi. Data cipherteks didekripsi pada ukuran blok 64 bit menjadi 64 bit data plainteks menggunakan 56 bit kunci internal. DES mentransformasikan input 64 bit dalam beberapa tahap dekripsi ke dalam output 64 bit. Pada proses dekripsi kunci yang digunakan pertama adalah kunci yang digunakan terakhir pada proses enkripsi dan kunci yang digunakan pertama pada proses enkripsi akan digunakan terakhir pada proses dekripsi.

3. Hasil Penelitian dan Pembahasan Implementasi program MATLAB

Implementasi program dilakukan pada program MATLAB. Didalam MATLAB ini telah menyediakan komponen yang berguna untuk mempermudah mendesain tampilan program. Sehingga dalam membuat program tidak diperlukan dalam penulisan kode untuk membuat tampilannya. Tampilan di dalam MATLAB dapat diatur dengan cara mengambil tools yang akan dipakai meletakan pada form yang sudah tersedia.

3.1 Pembangkitan Kunci Internal DES pada MATLAB

Enkripsi dan dekripsi DES memiliki 16 putaran, maka dibutuhkan kunci internal pada setiap putaran, untuk mendapatkan kunci internal pada setiap putaran dengan menggunakan 3 langkah, yaitu: permutasi PC-1, leftshift, permutasi PC-2.

1. Implementasi permutasi PC-1

```
function[vKey56] = fKeyPermutation(vKey64)
```

2. Implementasi *leftshift*

```
function[vKeyX28, vKeyY28] = fKeyShift(vKey56, iRoundNumber)
```

3. Implementasi permutasi PC-2

```
function[vKey56] = fKeyPermutation(vKey64)
```

3.2 Enkripsi dan Dekripsi Algoritma DES pada MATLAB

Algoritma enkripsi DES menggunakan 3 jenis tahapan, yaitu: Initial Permutation (IP) pada plaintext, Fungsi *f*, Invers Initial Permutation (IP^{-1}).

1. Implementasi Initial Permutation (IP) pada MATLAB

```
function[vNewValueLeft32, vNewValueRight32] =
fRound(iRoundNumber, vValueLeft32, vValueRight32, vKey64)
```

2. Implemtasi Expansion pada MATLAB

```
function[vValueRight48] = fFExpansionPermutation(vValueRight32)
```

3. Implementasi S-box pada MATLAB

```
function[vValueRight32] = fFSBoxSubstitution(vValueRight48)
```

4. Implementasi P-Box dengan matlab pada MATLAB

```
function[vNewValueRight32] = fFPBoxPermutation(vValueRight32)
```

5. Implementasi fungsi *f* pada MATLAB

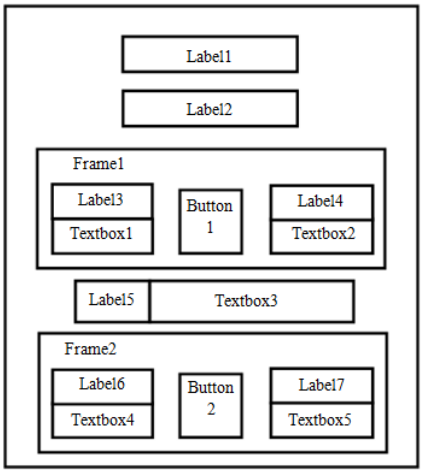
```
function[vValueRight32] = fF(vValueRight32, vKeyI48)
```

6. Implementasi Inverse Initial Permutation (IP^{-1}) pada MATLAB

```
function[vBlockPermuted64] = fFinalPermutation(vValueLeft32,
```

3.3 Rancangan View Program

Rancangan ini adalah tampilan yang akan muncul pertama kali ketika program DES dijalankan.

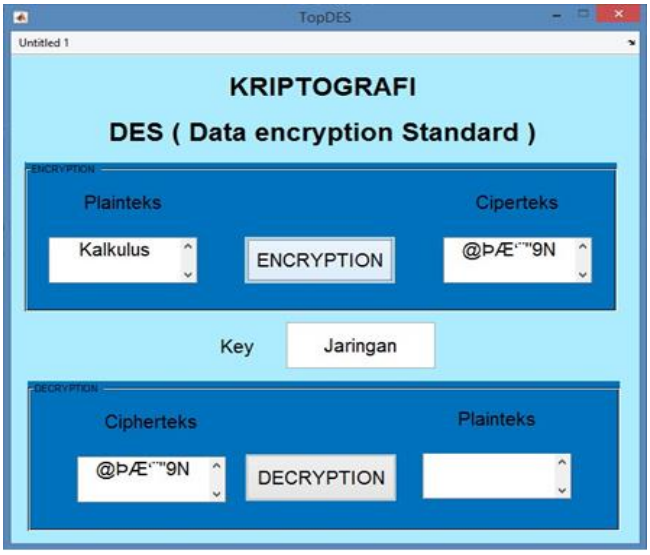


3.4 Implementasi Program

Implentasi view program akan melakukan pengprosesan enkripsi dan dekripsi pada Algoritma DES sebagai berikut:

a) Implementasi enkripsi

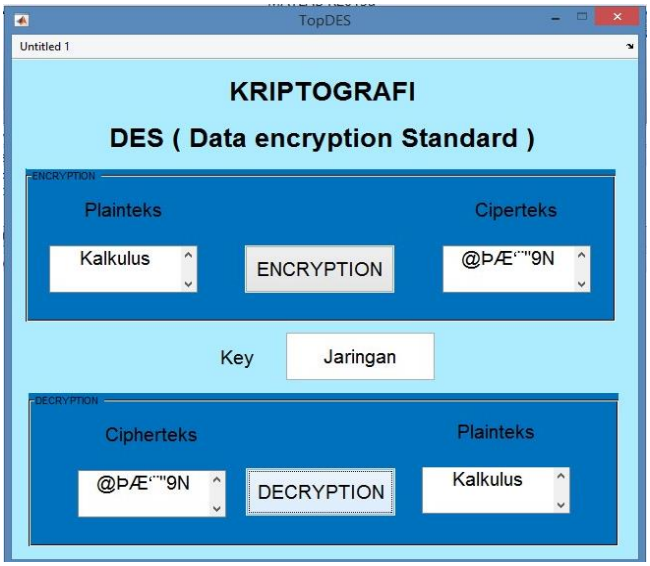
Apabila pengguna ingin menginput suatu plaintek : “Kalkulus” dengan Kunci: “Jaringan” menjadi chiperteks. Langkah pertama adalah menginputkan plainteks yang akan digunakan. Lalu menginputkan kunci yang akan digunakan untuk proses enkripsi dan dekripsi, selanjutnya menekan tombol “ENCRYPTION” yang akan mendapatkan cipherteks.



Gambar 3. Implementasi Enkripsi

b) Implementasi Dekripsi

Apabila pengguna ingin meng input suatu Chiperteks: @ÞÆ‘–“9N dengan Kunci : “Jaringan” menjadi plainteks. Langkah pertama adalah menginputkan plainteks yang akan digunakan. Lalu menginputkan kunci yang akan digunakan untuk proses enkripsi dan dekripsi, selanjutnya menekan tombol “DECRYPTION” yang akan mendapatkan plainteks kembali.



Gambar 4. Implementasi Dekripsi

4. Kesimpulan

Teknik keamanan algoritma *Data Encryption Standard* (DES), secara umum DES terbagi menjadi tiga kelompok, yaitu pemprosesan kunci ekternal 64 bit (sesuai ukuran blok), tetapi hanya 56 bit yang dipakai (8 bit terakhir tidak digunakan), enkripsi data 64 bit, dan dekripsi data 64 bit yang mana satu kelompok saling berinteraksi satu dengan yang lainnya yang berfungsi untuk menyediakan keamanan. Keamanan data dengan cara mengenkripsi data sehingga bagi orang yang tidak berhak tidak akan dapat membaca data tersebut tanpa memiliki kuncinya. Teknik ini sangat efektik karena dapat menjaga kerahasiaan data khususnya data dan juga memerlukan waktu yang sangat lama untuk dapat menemukan kunci yang benar.

Algoritma DES banyak melakukan oprasi permutasi dan substitusi dalam bentuk matriks sehingga, software MATLAB membantu proses enkripsi dan dekripsi DES dilaksanakan dengan cepat, tepat, dan efisien. *Graphical User Interface* (GUI) MATLAB digunakan untuk membangun interface dari algoritma kriptografi DES membantu menyembunyikan kerumitan dari algoritma DES. Fasilitas GUI yang disediakan MATLAB mempermudah pembangunan interface yang mudah digunakan oleh pengguna, karena algoritma DES dan interfacenya dibuat dalam bahasa pemrograman yang sama.

Dalam penulisan skripsi ini, pada *interface* GUI MATLAB sangat terbatas dengan fitur-fitur desain *interface* dan desain yang kaku. Harapan penulis dimasa yang akan datang dapat diimplementsikan selain pada vicual GUI yang dapat menyajikan desain interface yang lebih baik, desain interface yang lebih banyak dan kompetable dengan software MATLAB.

Referensi

[1] Munir, Rinaldi. 2006. *Kriptografi*. Bandung : Penerbit Informatika.

[2] Sadikin, Rifki. 2012. *Kriptografi Untuk Keamanan Jaringan*. Yogyakarta: Penerbit Andi.

[3] Stallings, Wiliams.2003. *Komputer Security And Cryptography*. New Jersey: A Jhon Wiley&Sons,Inc.

[4] E. Harahap, Failure prediction method for network management system by using Bayesian network and shared database, IEEE 8th Asia-Pacific Symposium on Information and Telecommunication Technologies (APSITT), Malaysia, 2010. pp. 1-6.

[5] Budi, prasetyo. 2013. [online] Tersedia: http://eprints.undip.ac.id/41211/1/Budi_Prasetyo.pdf [Diakses 16 Mei 2016].

[6] Supardi. 2010. [online] Tersedia: http://staff.uny.ac.id/sites/default/files/pendidikan/Supardi,%20M.Si/ pemograman%20MATLAB.pdf. [Diakses 16 Desember 2016].

[7] E. Harahap, J. Wijekoon, R. Tennekoon, F. Yamaguchi, S. Ishida. *Modeling of Router-based Request Redirection For Content Distribution Network.* International Journal of Computer Applications, volume 76, issue 13, pp. 37-46. New York 76.13 (2013). DOI: 10.5120/13310-0857

[8] A. Tulloh, Y. Permanasari, E. Harahap. 2016. *Kriptografi Advanced Encryption Standard (AES) Untuk Penyandian File Dokumen.* Jurnal Matematika UNISBA. Vol. 15 No 1, Mei 2016. pp. 7-14.

[9] Tennekoon, R., Wijekoon, J., Harahap, E., dan Nishi, H. 2014. *Per-hop data encryption protocol for transmitting data securely over public netrwork.* In Procedia Computer Science. Volume. 32. p. 965-972. Tersedia: DOI: 10.1016/i.procs.2014.05.519

[10] Y. Permanasari, E. Harahap. *Algoritma Data Encryption Standard (DES) Pada Electronic Code Book (ECB)*. Jurnal Matematika UNISBA. Vol 6 No 1, 2007. Pp. 77-84.

[11] R. Tennekoon, J. Wijekoon, E. Harahap, H. Nishi, E. Saito, S *Katsura. Per hop data encryption protocol for transmission of motion control data over public networks*. Proceeding Advanced Motion Control (AMC) on IEEE 13th International Workshop, Yokohama, Japan. 2014. pp.128-133.