Jurnal Coding, Sistem Komputer Untan
Volume 04, No.2 (2016), hal. 1-12 ISSN : 2338-493X

APLIKASI ENKRIPSI DAN DEKRIPSI UNTUK KEAMANAN DOKUMEN MENGGUNAKAN TRIPLE DES DENGAN MEMANFAATKAN USB FLASH DRIVE

[1]Joko Susanto, [2]Ilhamsyah, [3]Tedy Rismawan
[1] [3]Jurusan Sistem Komputer, Fakultas MIPA Universitas Tanjungpura
[2]Jurusan Sistem Informasi, Fakultas MIPA Universitas Tanjungpura
Jl. Prof. Dr. H. Hadari Nawawi, Pontianak
Telp./Fax.: (0561) 577963
e-mail:
[1]jokosusantosiskom@gmail.com, [2]ilhamsm99@gmail.com,
[3]tedyrismawan@siskom.untan.ac.id

Abstrak

Keamanan dan kerahasiaan data dan informasi pada jaringan komputer saat ini menjadi isu yang sangat penting dan terus berkembang. Beberapa kasus menyangkut keamanan data dan informasi saat ini menjadi suatu pekerjaan yang membutuhkan biaya penanganan dan pengamanan yang cukup besar. Sesuai dengan perkembangan zaman diperlukan suatu cara untuk mengamankan data dan informasi. Salah satu cara untuk mengamankan data dan informasi adalah dengan cara mengubah data dan informasi tersebut ke dalam bentuk data dan informasi yang tidak dapat dimengerti oleh pihak lain, yaitu dengan cara penyandian. Kriptografi adalah suatu metode keamanan untuk melindungi suatu informasi dengan menggunakan kata-kata sandi yang hanya bisa dimengerti oleh orang yang berhak mengakses informasi tersebut. Algoritma 3DES adalah salah satu jenis metode kriptografi dari algoritma pengembangan algoritma DES (Data Encryption Standard). Perbedaan DES dan 3DES terletak pada panjang kunci yang digunakan Pada DES menggunakan satu kunci yang panjangnya 56-bit, sedangkan pada 3DES menggunakan 3 kunci yang panjangnya 168-bit. Karena tingkat kerahasian algoritma 3DES terletak pada panjangnya kunci yang digunakan, maka penggunaan algoritma 3DES dianggap lebih aman dibandingkan dengan algoritma DES. Hasil dari algoritma Triple DES. Untuk memudahkan penggunaan algoritma 3DES, maka dibuat suatu program algooritma 3DES dengan alat bantu software komputer yang dapat mengenkripsi dan mendekripsi dokumen dan memanfaatkan USB Flash Drive sebagai salah satu kunci dalam penggunaan aplikasi 3DES.

Kata Kunci*: keamanan, DES, 3DES, USB Flash Drive.*

1. PENDAHULUAN

Salah satu hal penting dalam komunikasi menggunakan komputer di dalam jaringan adalah menjamin keamanan pesan, data ataupun informasi. Keamanan dan kerahasiaan data pada jaringan komputer saat ini menjadi isu yang sangat penting dan terus berkembang. Beberapa kasus menyangkut keamanan data saat ini menjadi suatu pekerjaan yang membutuhkan biaya penanganan dan pengamanan yang cukup besar.

Sesuai dengan perkembangan zaman diperlukan suatu cara untuk mengamankan data dan informasi. Salah satu cara untuk mengamankan data adalah dengan cara mengubah data tersebut kedalam bentuk data yang tidak dapat dimengerti oleh pihak lain, yaitu dengan cara penyandian. Usaha perlindungan data dapat dilakukan dengan berbagai cara, salah satunya dengan metode kriptografi. Usaha pengamanan dalam kriptografi dapat dilakukan dengan mengenkripsikan data ke dalam bentuk yang tidak dapat dikenali. Untuk mengembalikan menjadi seperti data semula, dapat dilakukan dengan mendekripsi data yang tidak dapat dikenali tersebut.

Dalam penelitian ini sistem yang akan dibangun adalah mengubah dokumen dalam bentuk yang tidak dapat dikenali agar kerahasiaannya dapat terjaga. Oleh karena itu salah satu metode yang digunakan untuk mengamankan dokumen ialah metode enkripsi dan dekripsi.

2. LANDASAN TEORI

2.1 Kriptografi

Kriptografi adalah suatu metode keamanan untuk melindungi suatu informasi dengan menggunakan kata-kata sandi yang hanya bisa dimengerti oleh orang yang berhak mengakses informasi tersebut. [1]

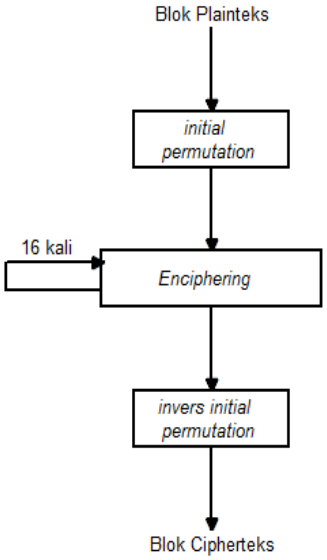
Kriptografi sebenarnya adalah metode yang sering sekali digunakan untuk melindungi berbagai macam data yang prosesnya disebut dengan *encryption*, yaitu suatu proses yang mengkonversikan sebuah pesan *plaintext* menjadi sebuah *ciphertext* yang bisa dibalikan ke bentuk asli seperti semula, yang juga disebut sebagai proses *decoding* atau *decryption* [2].

Dalam penelitian ini, enkripsi yang digunakan terhadap suatu dokumen dengan mengubahnya menjadi bentuk yang tidak dikenal. Setiap dokumen yang dienkripsikan memiliki kunci saat proses penyandian. Kemudian dokumen tersebut dapat didekripsikan atau dikembalikan seperti biasa dengan kunci yang dimiliki setiap dokumen yang bisa membuka sistem keamanan enkripsi tersebut, agar keamanan dokumen dapat terjaga kerahasiannya dari orang-orang yang tidak berwenang untuk mengetahui informasi dokumen tersebut.

2.1.1 Algoritma DES (*Data Encryption Standard*)

DES adalah algoritma *cipher* blok yang popular karena dijadikan standar algoritma enkripsi kunci simetris, meskipun saat ini standar tersebut telah digantikan dengan algoritma baru, AES, karena DES sudah tidak dianggap aman lagi. Sebenarnya DES adalah nama standar enkripsi simetris, nama algoritma – algoritma enkripsi sendiri adalah DEA (*Data Encryption Algoritma*), namun nama DES lebih popular daripada DEA. Algoritma DES dikembangkan di IBM dibawah kepimpinan W.L. Tuchman pada tahun 1972. Algoritma ini didasarkan pada algoritma *Lucifer* yang dibuat oleh Horst Feistel. Algoritma ini telah disetujui oleh *National Bureu of Standard* (NBS) setelah penilaian kekuatannya oleh *National Security Agency* (NSA) Amerika Serikat [3].

DES termasuk ke dalam sistem kriptografi simetri dan tergolong jenis *cipher* blok. DES beroperasi pada ukuran blok 64 bit. DES mengenkripsikan 64 bit plainteks menjadi 64 bit cipherteks dengan menggunakan 56 bit kunci internal (*internal key*) atau upa-kunci (*subkey*). Kunci intenal dibangkitkan dari kunci eksternal (*external key*) yang panjangnya 64 bit [3].



Gambar 1. Skema Global Algoritma DES [3]

Skema global dari algoritma DES sebagai berikut:

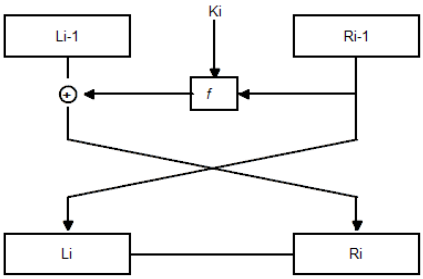
1. Blok plainteks dipermutasi dengan matriks permutasi awal (*initial permutation* atau IP).
2. Hasil permutasi awal kemudian di-*enciphering* sebanyak 16 kali (16 putaran). Setiap putaran menggunakan kunci internal yang berbeda.
3. Hasil *enciphering* kemudian dipermutasi dengan matriks permutasi balikan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

Di dalam proses *enciphering*, blok plainteks terbagi menjadi dua bagian yaitu kiri (L) dan kanan (R). Yang masing-masing panjangnya 32 bit. Kedua bagian ini masuk ke dalam 16 putaran DES. Pada setiap putaran *I*, blok R merupakan masukan untuk fungsi transformasi yang disebut *f*. Pada fungsi *f* di-XOR-kan dengan blok *L* untuk mendapatkan blok *R* yang baru. Sedangkan blok *L* yang baru langsung diambil dari blok *R* sebelumnya. Ini adalah satu putaran DES. Secara matematis, satu putaran DES dinyatakan sebagai persamaan [3].

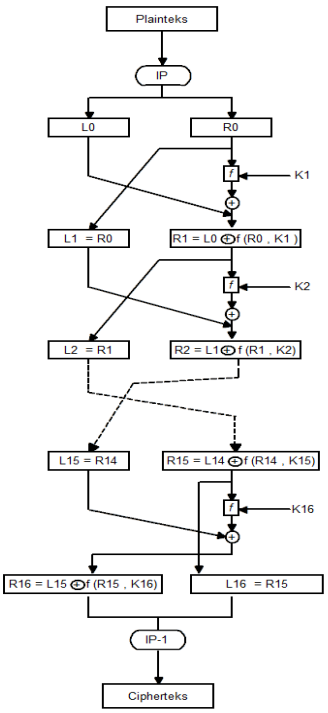
$$Li = Ri-1 \quad (1)$$

$$Ri = Li-1 \oplus f(Ri-1, Ki) \quad (2)$$

Satu putaran DES merupakan jaringan Feistel (lihat gambar 1). Perlu dicatat dari gambar 1 bahwa (L_{16}, R_{16}) merupakan keluaran dari putaran ke-16, maka (R_{16}, L_{16}) merupakan pra-cipherteks (*pre-ciphertext)* dari *enciphering* ini. Cipherteks yang sebenarnya diperoleh dengan melakukan permutasi awal balikan, IP^{-1}, terhadap blok pra-cipherteks.



Gambar 2. Jaringan Feistel untuk satu putaran DES [3]



Gambar 3. Algoritma Enkripsi DES [3]

2.1.2 Algoritma Triple DES

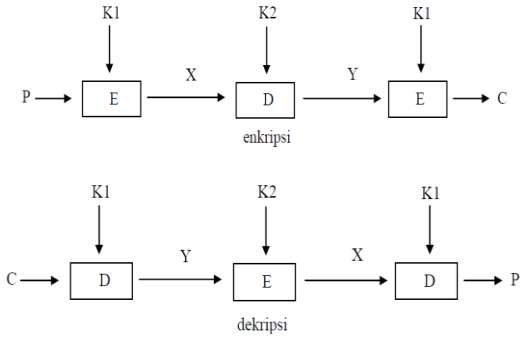
Triple DES adalah sebuah cipher yang dibentuk oleh DES menggunakan tiga kali. Penggunaan tiga langkah ini penting mencegah *meet-in-the-middle attack* sebagaimana pada *double DES*. Bentuk sederhana *Triple* DES adalah: [3]

Enkripsi : C = $Ek3(Ek2(Ek1(P)))$ (3)
Dekripsi : P = $Dk1(Dk2(Dk3(C)))$ (4)

Varian ini umum dikenal sebagai mode EEE (*Encrypt Encrypt Encrypt*) untuk enkripsi karena pada proses enkripsi semuanya menggunakan enkripsi. Untuk menyederhanakan *interoperability* antara DES dan Triple DES, maka langkah ditengah (pada proses enkripsi Tripel DES) diganti dengan dekripsi (mode EDE(*Encrypt Decrypt Encrypt*)). Dengan pengubahab ini, maka dibuat beberapa versi Triple DES. Versi pertama Triple DES mrnggunakan dua kunci, K1 dan K2: [3]

Enkripsi : C = $Ek1(Dk2(Ek1(P)))$ (5)
Dekripsi : P = $Dk1(Ek2(Dk1(C)))$ (6)

Enkripsi DES tunggal dengan kunci K dapat dinyatakan sebagai Triple DES-EDE dengan K1=K2=K. Memperlihatkan versi Triple DES yang menggunakan dua kunci. Penggunaan enkripsi pada langkah ditengah tidak mempengaruhi keamanan algoritma, dapat dilihat pada gambar 4. [3].

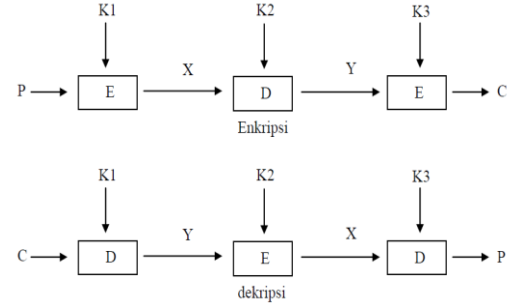


Gambar 4. Enkripsi dan Dekripsi Triple DES menggunakan 2 kunci [3]

Versi Triple DES yang kedua menggunakan tiga buah kunci, *K1*, *K2* dan *K3* sebagai berikut:

Enkripsi : C = $Ek3(Dk2(Ek1(P)))$ (7)
Dekripsi : P = $Dk1(Ek2(Dk3(C)))$ (8)

Proses enkripsi dan dekripsi ini dapat dilihat pada gamabar 5.



Gambar 5. Diagram Enkripsi dan Dekripsi Triple DES yang menggunakan tiga buah kunci [3]

2.2 USB Flash Drive

USB Flash Drive atau Flash Disk merupakan perangkat penyimpanan data yang terdiri dari memori flash dengan terintegrasi *interface Universal Serial Bus* (USB). USB flash drive bersifat *removable* dan dapat ditulis ulang (re-write), dan secara fisik ukurannya jauh lebih kecil daripada *floppy disk*, beratnya kurang dari 30 gr. Kapasitas penyimpanan UFD sebesar 2 terabyte telah direncanakan untuk diciptakan [4].

USB Flash Drive memiliki banyak kelebihan dibandingkan alat penyimpanan data lainnya. Pada kenyataannya, pemanfaatan flash drive telah berkembang untuk berbagai hal. Misalnya sebagai alat untuk *booting*. Beberapa aplikasi juga dapat dijalankan secara langsung dari flash drive tanpa harus meng-install-nya terlebih dahulu ke komputer. Sehingga disebut dengan aplikasi *portable*. Dan juga banyak aplikasi yang dapat memberi manfaat untuk mengunci komputer dengan menggunakan flash disk atau memberi password otomatis ke flash disk [4].

3. METODE PENELITIAN

Penelitian ini berjudul "Aplikasi Enkripsi dan Dekripsi untuk Keamanan Dokumen Menggunakan Triple DES dengan Memanfaatkan USB Flash Drive". Metode yang digunakan pada penelitian ini mencakup studi pustaka serta analisa kebutuhan yang akan dikumpulkan buat kebutuhan *software* setelah itu dilakukan desain untuk tampilan antarmuka lalu dilakukan suatu pengembangan untuk mengatasi masalah yang ada pada latar belakang dan terakhir dilakukan pengujian agar aplikasi yang dibuat berhasil atau gagal.

4. PERANCANGAN DAN IMPLEMENTASI

Aplikasi yang dibangun merupakan suatu aplikasi yang memproses enkripsi dan dekripsi dokumen dengan ekstansi doc, docx, xls, xlsx dan pdf. Pada aplikasi ini terdapat satu *user* yang melakukan enkripsi dan dekripsi. Aplikasi ini berbasis *desktop* yang dibangun dengan bahasa *basic* dengan menggunakan algoritma Triple DES dan memanfaatkan USB Flash Drive sebagai salah satu kunci. Prinsip kerja sistem secara keseluruhan dapat dilihat pada gambar 6.

Gambar 6. Arsitektur Sistem

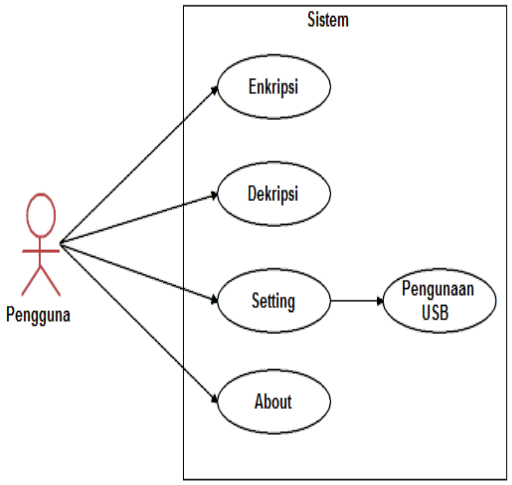
Penjelasan arsitektur Sistem dari gambar 6 ini ialah dimana terdapat suatu aplikasi keamanan dokumen dengan menggunakan bahasa *basic* dan algoritma kriptografi Triple DES sebagai metode. Aplikasi ini dibangun berbasis *desktop*. Proses enkripsi dan dekripsi oleh pengguna, dilakukan dengan cara memasukkan dokumen yang ingin dienkripsi dan dekripsi serta kunci yang akan dienkripsi dan dekripsikan setelah itu sistem akan melakukan proses enkripsi dan dekripsi yang telah dimasukkan pengguna.

4.1 Use Case Diagram

Dalam tahap perancangan digunakan *use-case diagram* untuk menggambarkan fungsionalitas sistem yang dilakukan oleh pengguna. Pengguna dapat melakukan beberapa proses saat memulai aplikasi. *Use-case diagram* sistem ditunjukan pada Gambar 7. Proses yang dilakukan pengguna adalah:

1. Pengguna mengenkripsi dokumen yang telah di *input* kunci pada aplikasi.

2. Pengguna mendekripsi dokumen yang telah di *input* kunci pada aplikasi.
3. Pengguna dapat mengatur kunci yang akan digunakan.
4. Pengguna dapat melihat info tentang aplikasi.



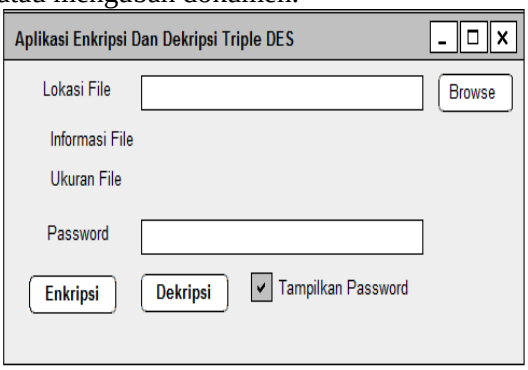
Gambar 7. *use-case diagram*

4.2 Perancangan Antarmuka

Untuk membuat aplikasi ini, diperlukan perancangan antarmuka yang merupakan penghubung antara sistem dan pengguna. Terdapat beberapa tampilan aplikasi yang akan dibuat diantaranya:

1) Rancangan tampilan Enkripsi dan Dekripsi

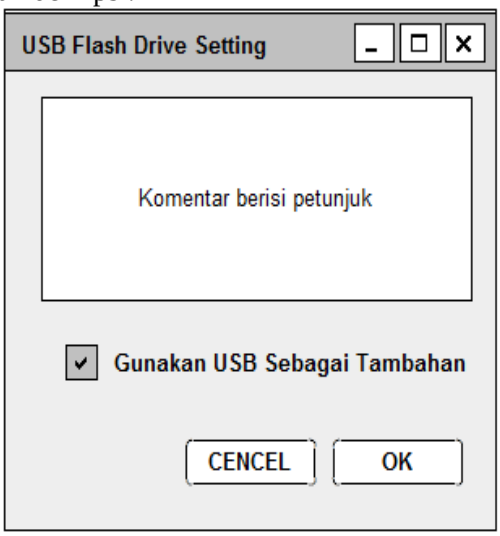
Rancangan tampilan enkripsi dan dekripsi ini dapat dilihat pada gambar 8. ketika pengguna (*user*) akan mengkonversi atau mengubah dokumen.



Gambar 8. Rancangan Tampilan Enkripsi dan Dekripsi

2) Rancangan pengaturan USB Flash drive

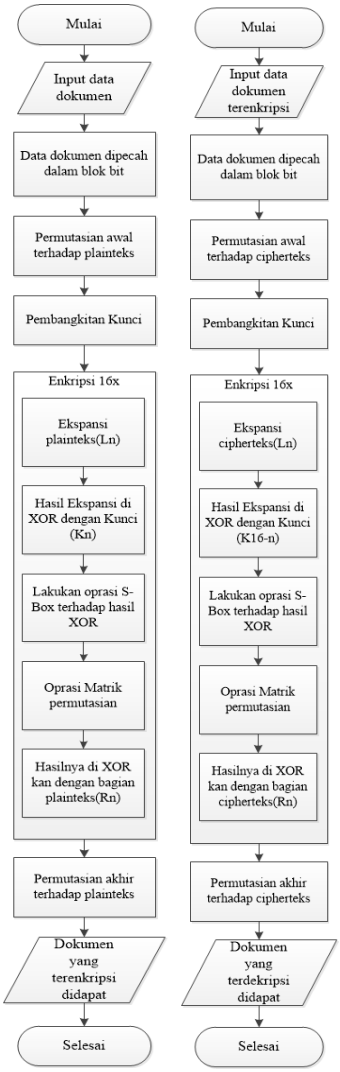
Rancangan tampilan pengaturan USB Flash Drive ini bisa dilihat pada gambar 9. yang digunakan ketika pengguna (*user*) akan mengkoneksikan atau menghubungkan USB Flash Drive sebagai salah satu kunci enkripsi dan dekripsi.



Gambar 9. Rancangan Tampilan Pengaturan USB Flash Drive

4.3 Perancangan Proses Enkripsi Dekripsi Dengan Triple DES

Proses enkripsi dan dekripsi dilakukan oleh *user* atau pengguna, dengan cara memasukkan dokumen yang ingin dienkripsi dan dekripsiserta kunci yang akan dienkripsi setelah itu sistem akan melakukan proses enkripsi dan dekripsi yang telah dimasukkan pengguna. Proses enkripsi dan dekripsi dilakukan dalam proses DES terlebih dahulu dengan mengubah dan memecah blok-bit plainteks (Ln dan Rn) dan juga mengubah dan membentuk kunci (Kn) sebanyak 16, setelah itu plainteks/cipherteks diubah dengan tabel permutasian awal (IP) lalu dilakukan proses enkripsi/dekripsi kunci dan plainteks/cipherteks sebanyak 5 tahap, yaitu tahap ekspansi, tahap XOR kunci dengan plainteks, tahap subsitusi *box*, tahap permutasian *box* dan terakhir XOR-kan hasil dengan bagian plainteks/cipherteks yang dipecah. Proses ini dilakukan sebanyak 16 kali sesuai dengan pecahan kunci yang dipecah sebanyak 16. Terakhir dilakukan proses permutasian akhir (IP-1), setelah itu didapat plainteks/cipherteks yang dienkripsi dan dekripsi. Dalam proses Triple DES terdapat proses DES yang dilakukan sebanyak 3 kali. Adapun *flowchart* enkripsi dan dekripsi Triple DES dapat dilihat pada gambar 10.



Gambar 10. *flowchart* enkripsi dan dekripsi

4.4. Penggunaan Algoritma Triple DES

Plainteks (x) = jokosusa
Kunci (K) = muhammad

1. Langkah pertama

Ubah plainteks ke ASCII

j = 106
o = 111
k = 107
o = 111
s = 115
u = 117
s = 115
a = 97

Ubah plainteks dari ACSII ke bentuk biner.

j = 0110 1010
o = 0110 1111
k = 0110 1011
o = 0110 1111
s = 0111 0011
u = 0111 0101
s = 0111 0011
a = 0110 0001

Ubah kunci ke ASCII

m = 109
u = 117
h = 104
a = 97
m = 109
m = 109
a = 97
d = 100

Ubah Kunci dari ASCII ke bentuk biner.

m = 0110 1101
u = 0111 0101
h = 0110 1000
a = 0110 0001
m = 0110 1101
m = 0110 1101
a = 0110 0001
d = 0110 0100

2. Langkah kedua

Lakukan pengurutan plainteks yang bilangan biner dari 1 - 8 bit dari huruf "j" sampai bit akhir 57 – 64 dari huruf "a" menjadi sebanyak 64 bit. Pengurutan bit plainteks dapat dilihat pada tabel plainteks berikut:

Tabel 1. Plainteks

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

Setelah itu lakukan operasi *initial permutation* (IP) pada bit plainteks menggunakan tabel IP berikut:

Tabel 2. *initial permutation* (IP)

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

Urutan bit plainteks ke 58 diletakkan pada posisi 1, urutan plainteks ke 50 diletakkan di posisi 2, urutan bit pada plainteks urutan ke 42 diletakkan di di posisi

3, dan seterusnya. Sehinggan hasil *output*-nya adalah 11111111 01110000 00101010 11111110 00000000 11111111 00001111 01011111. Pecahan bit pada IP di pecah menjadi 2 bagian yaitu:
Hasil plainteks = 11111111 01110000 00101010 11111110 = L0, 00000000 11111111 00001111 01011111 = R0

3. Langkah ketiga

Lakukan pengurutan kunci yang bilangan biner dari 1 - 8 bit dari huruf "m" sampai bit akhir 57 – 64 dari huruf "d" menjadi sebanyak 64 bit. Pengurutan bit kunci dapat dilihat pada tabel kunci berikut:

Tabel 3. Kunci

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

Mengubah kunci yang akan digunakan untuk mengenkripsi plainteks dengan menggunakan tabel permutasi kompresi (PC-1), pada langkah ini terjadi kompresi dengan membuang 1 bit masing-masing blok dari 64 bit menjadi 56 bit.

Tabel 4. PC-1

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

Dapat dilihat pada tabel 4.6 tidak terdapat urutan bit 8, 16, 24, 32, 40, 48, 56, dan 64 karena tidak di kompresi. Berikut ini hasil *output* dari proses permutasi kompresi (PC-1) 00000000111111 11111111110000 00000000101100 11001101010010

Pecah bit pada PC-1 menjadi 2 bagian kiri dan kanan, sehinggan menjadi:

C0 = 00000000111111 11111111110000, D0 = 00000000101100 11001101010010.

4. Langkah keempat

Lakukan pergeseran kiri (*Left Shift*) pada C0 dan D0, sebanyak 1 atau 2 kali berdasarkan kali putaran yang ada pada tabel putaran sebagai berikut:

Tabel 5. *Left Shift*

Putaran, *i*	Jumlah pergeseran bit
1	1
2	1
3	2
4	2
5	2
6	2
7	2
8	2
9	1
10	2
11	2
12	2
13	2
14	2
15	2
16	1

Untuk putaran ke 1 dilakukan pergeseran 1 bit ke kiri, untuk putaran ke 2 dilakukan pergeseran 1 bit ke kiri, untuk putaran ke 3 dilakukan pergeseran 2 bit ke kiri, dan seterusnya. Berikut hasil *output* dari tabel 5.

Co = 00000000111111 11111111110000
Do = 00000000101100 11001101010010
Digeser 1 bit ke kiri
C1 = 0000000111111 111111111100000
D1 = 0000000101100 110011010100100
Digeser 1 bit ke kiri
C2 = 000000111111 1111111111000000
D2 = 000000101100 1100110101001000
Digeser 2 bit ke kiri
C3 = 0000111111 111111111100000000
D3 = 0000101100 110011010100100000
Digeser 2 bit ke kiri
C4 = 00111111 11111111110000000000
D4 = 00101100 11001101010010000000
Digeser 2 bit ke kiri
C5 = 111111 111111111110000000000
D5 = 101100 110011010100100000000
Digeser 2 bit ke kiri
C6 = 1111 111111111100000000000011
D6 = 1100 110011010100100000000010
Digeser 2 bit ke kiri
C7 = 111111111111000000000001111
D7 = 001100110101001000000001011
Digeser 2 bit ke kiri
C8 = 111111111100000000000111111
D8 = 110011010100100000000101100
Digeser 1 bit ke kiri
C9 = 111111111000000000000111111
D9 = 100110101001000000001011001
Digeser 2 bit ke kiri
C10 =111111100000000000111111111
D10 =011010100100000000101100110
Digeser 2 bit ke kiri
C11 =111110000000000011111111111
D11=101010010000000010110011001

Digeser 2 bit ke kiri
C12 = 11100000000000001111111111111
D12 = 10100100000000010110011001100
Digeser 2 bit ke kiri
C13 = 10000000000000111111111111111
D13 = 10010000000001011001100110100
Digeser 2 bit ke kiri
C14 = 00000000000011111111111111110
D14 = 01000000000101100110011010100
Digeser 2 bit ke kiri
C15 = 00000000011111111111111111000
D15 = 00000000101100110011010100100
Digeser 1 bit ke kiri
C16 = 00000000111111111111111110000
D16 = 00000000101100110011010100100

Setiap hasil putaran digabungkan kembali menjadi CiDi dan dimasukkan ke dalam tabel permutasi kompresi 2 (PC-2) dan terjadi kompresi data CiDi 56 bit menjadi CiDi 48 bit.

Tabel 6. 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

Dapat dilihat pada tabel 4.10 tidak terdapat urutan bit 9, 18, 22, 25, 35, 38, 43, dan 54 karena tidak di kompresi. Berikut hasil *output* Proses permutasi kompresi 2 (PC-2).

K1 = 111000 001011 111001 100110 000010 001000 011000 110100

K2 = 111000 001011 011001 110110 110100 000101 000001 111000

K3 = 111001 001101 011001 110110 010000 011011 101000 100100

K4 = 111001 101101 001101 110010 101100 000010 110010 110000

K5 = 101011 101101 001101 110011 001010 010000 101000 010111

K6 = 101011 110101 001101 011011 000101 110110 000010 010010

K7 = 001011 110101 001111 011001 001001 010001 000101 000101

K8 = 000111 110101 100111 011001 100000 101010 000011 000110

K9 = 000111 110100 100111 011001 010100 010010 000101 001011

K10 = 000111 110110 100110 011100 001001 101011 000000 001000

K11 = 000111 110010 110110 001101 011000 000011 010101 100110

K12 = 010110 110010 110010 101101 001011 001000 100010 101010

K13 = 110110 011010 110010 101100 010001 000101 110001 010011

K14 = 110100 001010 111010 101110 000011 111000 000001 111000

K15 = 111100 001011 111000 100110 100000 011101 110101 000000

K16 = 111100 001011 111000 100110 010111 010001 101000 001000

5. Langkah kelima

Langkah ini merupakan langkah enkripsi yang dilakukan setelah permutasi awal, setiap blok plainteks mengalami 16 kali putaran. Setiap putaran merupakan jaringan *faistel* yang secara matematis dinyatakan sebgai persamaan.

$Li = Ri-1$ (9)

$Ri = Li-1 \oplus f(Ri-1, Ki)$ (10)

K1 = 111000 001011 111001 100110 000010 001000 011000 110100 => 48 Bit

L0 = 11111111 01110000 00101010 11111110 => 32 Bit

L1 = R0 = 00000000 11111111 00001111 01011111 => 32 Bit

R1 = L0 + *f* (R0,K1)

Pertama, kita mengekspansi R0 dari 32 bit menjadi 48 bit dengan menggunakan tabel ekspansi.

Tabel 7. Tabel Ekspansi (E)

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

E(R0) = 100000 000001 011111 111110 100001 011110 101011 111110

Kedua, lakukan operasi XOR pada E(R0) dengan kunnci (K1).

K1 = 111000 001011 111001 100110 000010 001000 011000 110100 XOR E(R0) = 100000 000001 011111 111110 100001 011110 101011 111110

E(R0) + K1 = 011000 001010 100110 011000 100011 010110 110011 001010

Ketiga, lakukan operasi *S-box* (*Substitution Box*), dimana setiap group terdiri dari 6 bit.

E(R0) + K1 = *S1(B1), S2(B2), S3(B3), S4(B4), S5(B5), S6(B6), S7(B7), S8(B8)*

Jika B adalaha 6 bit blok, Bit pertama dan terakhir dari *B* mewakili bilangan desimal 0-3 (00-11). Anggap bilangan ini sebagai *i*. 4 bit tengah lainnya mewakilli bilangan desimal 0-15 (0000-1111) dan anggap ini sebagai *j*. Dengan demikian kita dapat melihat *i* sebagai baris dan *j* sebagai kolom. Sebagai contoh jika masukkan B = 011000, maka *i* = 00 dan *j* = 1100. Pada baris 0 dari kolom 12 didapat nilai 5, jadi keluarannya 0101 Dengan *Si(Bi)* menunjukan keluaran dari S-box ke-1. Hasil *output* dari proses *S-box* yang terdiri dari 4 bit setiap dari keluaran adalah 0101 0010 1011 1011 1000 0100 0101 0010.

Keempat , hasil dari proses *S-box* akan dimasukkan ke dalam opersi tabel *P-box*.

Tabel 8. *P-box*

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

(R0, K1) = 1100 0111 0101 0010 1000 0001 0100 1110

Kelima, mendapatkan nilai R1

L0 = 1111 1111 0111 0000 0010 1010 1111 1110 XOR (R0, K1) = 1100 0111 0101 0010 1000 0001 0100 1110

R1 = 0011 1000 0010 0010 1010 1011 1011 0000

Lakukan operasi enkripsi ini sebanyak 16 kali putaran, sehingga menghasilkan

R16 = 0111 1000 1100 1010 1100 1000 1110 1101, L16 = 0100 1001 1100 1001 0000 0010 0101 1110

6. Langkah keenam

Langkah terakhir adalah langkah menggabungkan R16 dengan L16 kemudian dipermutasikan untuk yang terakhir kali dengan tabel *invers initial permutation.*

Tabel 9. Permutasian akhir

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

Sehingga *input* dari proses permutasian akhir sebagai berikut:

R16L16 = 0111 1000 1100 1010 1100 1000 1110 1101 0100 1001 1100 1001 0000 0010 0101 1110.

Menghasilkan *output*:

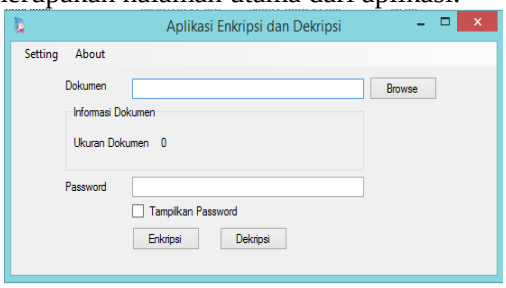
Cipherteks (dalam biner) = 10100001 00011010 00000011 11110111 11000010 01000001 11110111 00110101

Cipherteks yang didapat merupakan operasi DES pertama. Selanjutnya lakukan operasi DES lagi dengan kunci yang lain atau dengan kunci yang sama, operasi ini dilakukan sebanyak tiga kali dan didapat hasil kebentuk ASCII dari Triple DES adalah 121 129 81 232 136 221 90 21.

5. PENGUJIAN

5.1. Halaman Utama Aplikasi

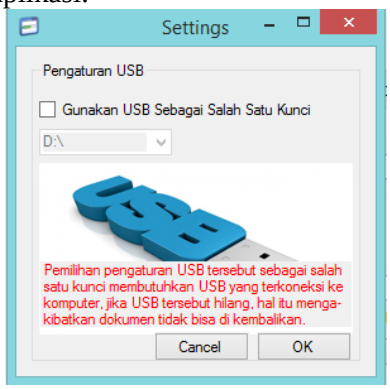
Ketika aplikasi dibuka, maka pengguna dapat melakukan enkripsi dan dekrispsi. Tampilan menu enkripsi dan dekripsi dapat dilihat pada gambar 11 yang merupakan tampilan saat dokumen diubah dan merupakan halaman utama dari aplikasi.



Gambar 11. Tampilan menu utama

5.2. Halaman Pengaturan USB Sebagai Kunci

Tampilan menu pengaturan dapat dilihat pada gambar 12 yang merupakan tampilan saat memilih USB sebagai salah satu kunci pada aplikasi.

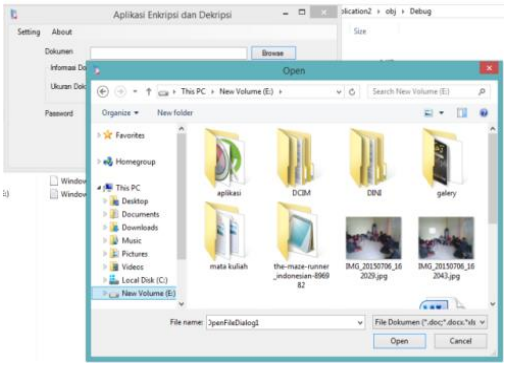


Gambar 12. Tampilan menu setting

5.1. Pengujian Enkripsi Dokumen

Untuk menggunakan aplikasi enkripsi dokumen, maka pengguna harus menekan tombol "*browse*" kemudian akan tampil pencarian dokumen yang ingin di enkripsi.

Tampilan pencarian dokumen yang ingin dienkripsi dapat dilihat pada gambar 13.

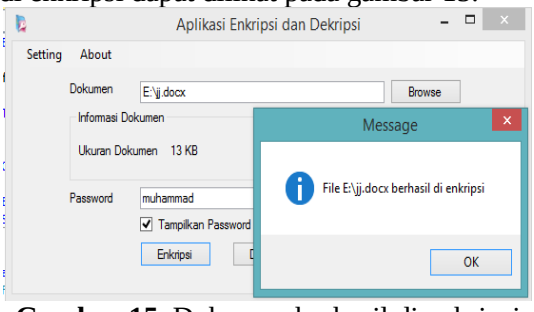


Gambar 13. Pencarian dokumen

Dalam pengujian aplikasi enkripsi dan dekripsi, dokumen yang akan dipilih dapat dilihat pada gambar 14 yang merupakan dokumen berekstensi docx yang berisikan kalimat "joko susanto" dengan *password* "Muhammad".

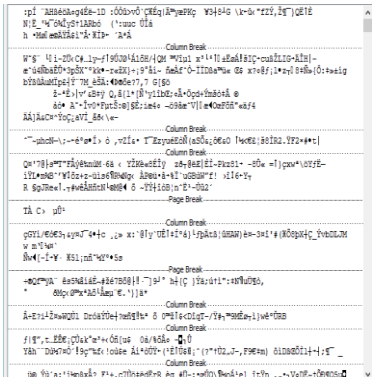
Gambar 14. Dokumen awal

Setelah dilakukan pemilihan dokumen, maka pengguna dapat memasukkan *password* dan menekan tombol enkripsi pada menu utama. Tampilan dari dokumen yang berhasil di enkripsi dapat dilihat pada gambar 15.



Gambar 15. Dokumen berhasil di enkripsi

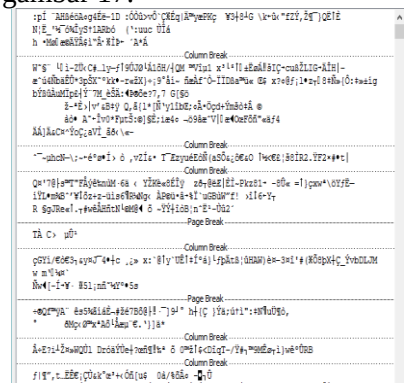
Hasil dari proses yang berhasil dienkripsi ialah berupa dokumen word yang isi tidak dapat dikenali. Tampilan dari dokumen yang berhasil di enkripsi dan tidak dapat dikenali dapat dilihat pada gambar 16.



Gambar 16. Hasil enkripsi

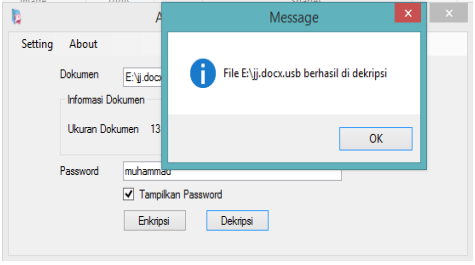
5.2. Pengujian Dekripsi Dokumen

Dalam pengujian aplikasi enkripsi dan dekripsi, dokumen yang akan dipilih ialah dokumen berekstensi docx yang telah terdekripsi berisikan kalimat yang tidak dapat dikenali dengan *password* "muhammad". Dokumen yang akan dipilih dapat dilihat pada gambar 17.



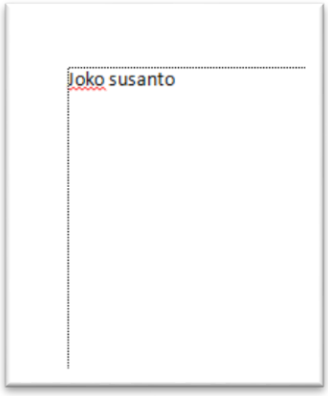
Gambar 17. Dokumen awal

Untuk membaca dokumen yang terdekripsi, perlu adanya pengembalian dengan aplikasi, maka pengguna harus menekan tombol "*browse*" kemudian akan tampil pencarian dokumen yang ingin dikembalikan atau yang ingin didekripsi, untuk tampilannya dapat dilihat pada gambar 13. Kemudian pengguna memasukkan *password* dari dokumen tersebut pada kolom *password* dan menekan tombol dekripsi. Tampilan dari dokumen yang berhasil didekripsi dapat dilihat pada gambar 18.



Gambar 18. Dokumen berhasil di dekripsi

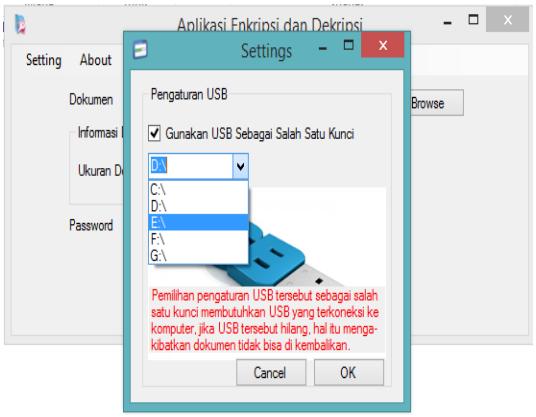
Hasil dari proses yang berhasil didekripsi ialah berupa dokumen *word* yang isi dapat dikenali atau dapat dibaca artinya. Tampilan dari dokumen yang berhasil di enkripsi dan tidak dapat dikenali dapat dilihat pada gambar 19.



Gambar 19. Hasil dekripsi

5.3. Pengujian Pengaturan Kunci

Dalam proses enkripsi dan dekripsi dokumen pada aplikasi Triple DES diperlukan adanya suatu kunci, dimana salah satu kunci aplikasi Triple DES memanfaatkan USB Flash Drive. Untuk menggunakan pengaturannya, pengguna memilih menu *setting* lalu pilih "pengaturan usb". Setelah itu pengguna memilih USB yang digunakan. Tampilan dari proses pemilihan kunci dapat dilihat pada gambar 20.



Gambar 20. Pengaturan USB

5.4. Hasil Pengujian

Berdasarkan hasil dari pengujian dengan semua uji kasus seperti yang diperlihatkan pada gambar pengujian 14, 16, 19 dan 20, didapat sebuah tabel pengujian terhadap tipe dokumen. Hasil pengujian dapat dilihat pada tabel 10.

Tabel 10. Tabel pengujian

No	FILE	UKURAN FILE	HASIL UKURAN FILE		KET
			Enkripsi	Dekripsi	
1.	coba.docx	15 KB	15 KB	15 KB	Berhasil
2.	algen.docx	73 KB	74 KB	73 KB	Berhasil
3.	konversi.docx	163 KB	163 KB	163 KB	Berhasil
4.	tabel.docx	35 KB	35 KB	35 KB	Berhasil
5.	coba2.xlsx	9 KB	9 KB	9 KB	Berhasil
6.	daftar.xlsx	12 KB	12 KB	12 KB	Berhasil
7.	data nilai.xlsx	34 KB	34 KB	34 KB	Berhasil
8.	transkrip.xlsx	21 KB	21 KB	21 KB	Berhasil
9.	coba3.pdf	84 KB	84 KB	84 KB	Berhasil
10.	proses.pdf	63 KB	63 KB	63 KB	Berhasil

Berdasarkan pada tabel 10 diperoleh hasil pengujian terhadap aplikasi enkripsi dan dekripsi untuk keamanan dokumen menggunakan Triple DES dengan memanfaatkan USB Flash Drive sesuai dengan spesifikasi pengujian yang telah ditentukan dan untuk semua uji kasus yang telah dilakukan. Adapun presentase keberhasilan sistem berdasarkan pada tebel 10 adalah:

$$\frac{\text{jumlah data uji berhasil}}{\text{jumlah data uji}} \text{ x } 100\% =$$

$$\frac{10}{10} \text{ x } 100\% = 100\%.$$

Akan tetapi tidak menutup kemungkinan dapat terjadi kesalahan suatu saat, pada aplikasi digunakan. Sehingga membutuhkan proses *maintenance* (pemeliharaan) untuk lebih mengetahui kekurangan dari aplikasi.

6. PENUTUP

6.1. Kesimpulan

Dengan dibangunnya "aplikasi enkripsi dan dekripsi untuk keamanan dokumen menggunakan Triple DES dengan memanfaatkan USB Flash Drive", maka diperoleh kesimpulan sebagai berikut:

1. Pada penelitian ini telah berhasil diuji aplikasi enkripsi dan dekripsi dengan algoritma triple DES dengan memanfaatkan USB Flash Drive dalam menjaga dokumen dengan ekstensi doc, docx, xls, xlsx dan pdf.
2. Berdasarkan hasil analisis metode kriptografi Triple DES dapat disimpulkan bahwa USB Flash Drive beroperasi dengan baik dalam mengamankan dokumen sebagai kunci.
3. Berdasarkan pengujian perangkat lunak yang dilakukan, maka diperoleh presentase keberhasilan sistem sebesar 100%.

6.2. Saran

Adapun saran terhadap sistem yang telah dibangun sebagai berikut:

1. Aplikasi enkripsi dan dekripsi menggunakan Triple DES ini sebaiknya dikembangkan tidak dalam bentuk dokumen yaitu doc, docx, xlc dan pdf akan tetapi mencakup seluruh data atau *file* seperti gambar, *video* dan lain-lain sehingga dapat bermanfaat untuk banyak pengguna.
2. Hasil enkripsi dan dekripsi ini sebaiknya dibandingkan dengan algoritma lain yang bertujuan untuk mengetahui apakah perbedaan dari segi ukuran *file* atau waktu prosesnya, baik simetris maupun asimetris misalnya dengan algoritma AES, RSA, Knapsack dan lain-lain.

DAFTAR PUSTAKA

[1] Hendrayanto, R. 2012. Program Aplikasi Enkripsi Dan Dekripsi SMS Pada Ponsel Berbasis Android Dengan Algoritma DES. *Jurnal Sistem Informatika Universitas Gunadarma Jawa Barat, VOL. 7, NO. 98, Halaman 631-633*

[2] Ariyus. 2008. *Pengantar Ilmu Kriptografi: Teori, Analisa, dan implementasi*. Yogyakarta: Penerbit Andi

[3] Reza. 2014. Analisa Enkripsi dan Dekripsi File Teks Menggunakan Algoritma Triple DES. *Teknik Informatika Universitas Islam Negri Sunan Gunung Djati bandung*.

[4] Rizki, 2013. Membuat USB key dan Generator Sebagai Kunci Aplikasi Menggunakan Visual Studio 2010. *Teknik Informatika STMIK AMIKOM Yogyakarta.*

[5] Primata, R. 2011. Penerapan Enkripsi Dan Dekripsi File Algoritma *Data Encryption Standard* (DES), *Jurnal Sistem Informasi Universitas Sriwijaya, VOL. 3, NO. 2, Halaman 371-387.*