ISSN 1693-1629
Edisi 5

Jurnal Ilmiah

ilmu komputer

Publikasi Hasil Penelitian Yang Diterbitkan Departemen Ilmu Komputer
Institut Pertanian Bogor

Vol.3 no.2 / Oktober 2005

Jurnal Ilmiah **ilmu komputer**

Diterbitkan oleh: Departemen Ilmu Komputer
Fakultas Matematika dan Ilmu Pengetahuan Alam - Institut Pertanian Bogor

Edisi 5 / Vol. 3. No. 2 Oktober 2005

ISSN : 1693-1629. Tanggal 4 April 2003

Susunan Redaksi

Sekretariat Jurnal Ilmiah **ilmu komputer :**

Departemen Ilmu Komputer FMIPA IPB
Jln. Raya Pajajaran, Kampus Baranangsiang Bogor 16144
Telp/Fax : 0251-356653 , E-mail : jurnal@ilkom.fmipa.ipb.ac.id
Rekening : Tabungan Taplus BNI Pajajaran Bogor.
No: **3031184** a.n.: *Annisa/Jurnal Ilkom*

Jurnal Ilmiah **ilmu komputer** *diterbitkan dua kali setahun, memuat tulisan ilmiah yang berhubungan dengan* ***bidang Ilmu Komputer****. Jurnal ini merupakan media publikasi ilmiah dan menerima tulisan dari luar IPB, berupa hasil penelitian atau bahasan tentang metodologi.*

Pihak perorangan / alumni yang telah memperoleh Jurnal Ilmu Komputer mohon mengganti biaya cetak Rp.50.000,-/expl, ditransfer melalui Tabungan Taplus BNI Pajajaran Bogor. No.Rek : 3031184 a.n.: Annisa / Jurnal Ilkom.

Daftar Isi

Analisis Algoritma *Triple*-DES untuk Penyandian Pesan

(Triple-DES Algorithm Analysis for Message Disguising)

Sony Hartono Wijaya[1], Sugi Guritman[2] dan Wisnu Ananta Kusuma[1]

[1] *Staf Pengajar Departemen Ilmu Komputer, FMIPA IPB*
[2] *Staf Pengajar Departemen Matematika, FMIPA IPB*

Abstrak

Paper ini mendiskusikan hasil analisis terhadap algoritma triple-DES sebagai varian dari DES (Data Encryption Standard) yang lebih kuat dan mampu melindungi informasi dengan baik. Analisis yang dilakukan meliputi analisis algoritma, analisis keamanan dan analisis hasil implementasi (kecepatan). Analisis algoritma terbagi menjadi dua yaitu analisis algoritma enkripsi dan analisis algoritma dekripsi.

Triple-DES menggunakan algoritma DES sebagai algoritma utama. Triple-DES dikembangkan untuk mengatasi kelemahan ukuran kunci yang digunakan pada proses enkripsi-dekripsi DES sehingga teknik kriptografi ini lebih tahan terhadap exhaustive key search yang dilakukan oleh kriptoanalis. Penggunaan triple-DES dengan suatu kunci tidak akan menghasilkan pemetaan yang sama seperti yang dihasilkan oleh DES dengan kunci tertentu. Hal itu disebabkan oleh sifat DES yang tidak tertutup (not closed). Sedangkan dari hasil implementasi dengan menggunakan modus Electronic CodeBook (ECB) menunjukkan bahwa walaupun memiliki kompleksitas/notasi-O yang sama (O(n)), proses enkripsi-dekripsi pada DES lebih cepat dibandingkan dengan triple-DES.

***Kata kunci**: Data Encryption Standard (DES), triple-DES, kriptografi, exhaustive key search, Electronic CodeBook (ECB), enkripsi, dekripsi.*

PENDAHULUAN

Latar Belakang

Salah satu contoh teknik kriptografi yang terkenal dalam sejarah adalah DES *(Data Encryption Standard)*. *DES* merupakan hasil kerja *Feistel* di *IBM* pada tahun 1977. Teknik ini banyak digunakan sebagai alat keamanan komersial di berbagai institusi dunia karena mampu melindungi informasi dengan baik.

Perkembangan teknologi dan teknik kriptografi menyebabkan *DES* rentan terhadap serangan. Hal itu disebabkan oleh panjang kunci yang digunakan oleh teknik ini terlalu pendek sehingga memungkinkan kriptoanalis menggunakan *exhaustive key search (brute-force attack)* untuk melakukan serangan. Dengan menggunakan *DES-cracker*, sebuah informasi yang terenkripsi dapat ditentukan *plaintext*nya dalam waktu kurang dari tiga hari dan biaya sebesar $ 250.000. Terlebih lagi jika dana yang dialokasikan untuk membangun sistem itu sebesar $ 1 juta, maka pesan terenkripsi tersebut dapat dipecahkan hanya dalam waktu 3,5 jam saja[1]. Oleh karena itu, varian dari *DES* yang lebih kuat dan mampu melindungi informasi dengan baik mulai dikembangkan. Salah satu varian dari *DES* yang kemudian dijadikan standar enkripsi dunia sejak tahun 1999 adalah *triple-DES*.

Tujuan

Tujuan dari penelitian ini adalah:

1. Mempelajari dan menganalisa kinerja algoritma *triple-DES*.
2. Mengimplementasikan algoritma *DES*, *double-DES* dan *triple-DES* ke dalam suatu sistem komputer dengan menggunakan bahasa pemrograman Microsoft Visual Basic.

Ruang Lingkup

Implementasi dengan menggunakan modus *Electronic CodeBook (ECB)* ditujukan untuk menyandikan pesan dalam *format file teks (.txt)*. Sedangkan analisis yang dilakukan meliputi analisis algoritma, analisis keamanan dan analisis hasil implementasi.

TINJAUAN PUSTAKA

Enkripsi, Dekripsi dan Kunci (Key)

Menurut *Schneier (1996)*, *enkripsi (E)* merupakan proses mengubah *plaintext* menjadi *ciphertext*. Sedangkan dekripsi *(D)* merupakan proses mengembalikan *ciphertext* menjadi *plaintext*.

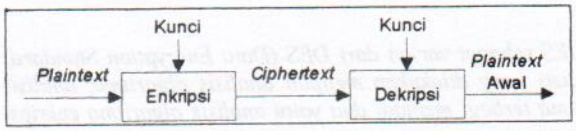
[1] http://www.tropsoft.com/strongenc/des3.htm [11 Desember 2003]

Semua teknik kriptografi menggunakan kunci untuk menjamin keamanan informasi. Kunci *(k)* merupakan salah satu anggota dari ruang kunci *(K)*. *DES* menggunakan kunci berukuran 64-bit, namun kunci yang efektif digunakan hanya 56-bit.

Proses enkripsi-dekripsi dengan menggunakan suatu kunci secara matematik dituliskan dengan:

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

Ilustrasi proses enkripsi–dekripsi dengan menggunakan kunci dapat dilihat pada **Gambar 1.**

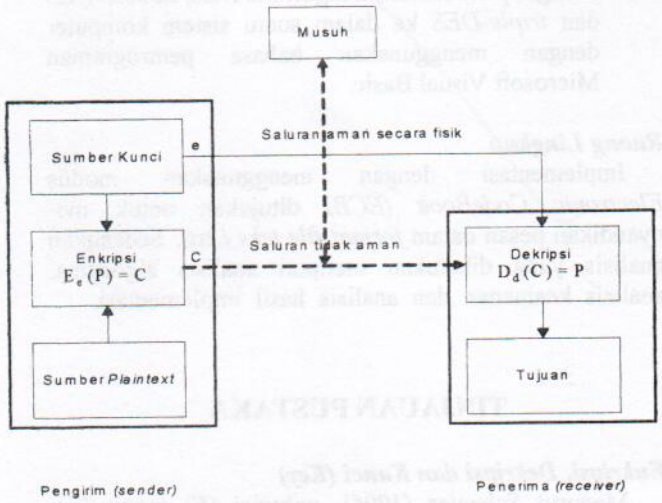


Gambar 1. *Proses enkripsi – dekripsi dengan menggunakan kunci.*

Algoritma Simetrik

Suatu skema enkripsi disebut sebagai enkripsi kunci simetrik jika untuk setiap pasangan kunci enkripsi dan dekripsi *(e,d)*, maka secara komputasi d *"mudah"* dihitung apabila e diketahui, dan sebaliknya *(Menezes, 1996)*. *Triple-DES* merupakan salah satu contoh algoritma simetrik. Skema enkrisi dua partai menggunakan kunci simetrik dapat dilihat pada **Gambar 2.**

Algoritma simetrik dibedakan menjadi dua, yaitu algoritma alir *(stream cipher)* dan algoritma blok *(block cipher)*. *Triple-DES* merupakan teknik kriptografi yang termasuk pada algoritma blok. Ukuran blok yang digunakan pada *triple-DES* adalah 64-bit.



Gambar 2. *Komunikasi dua partai menggunakan enkripsi kunci simetrik (Menezes, 1996).*

Tabel S-Box

Tabel *S-box* merupakan tabel yang digunakan untuk substitusi sederhana, yaitu pemetaan m-bit *input* menjadi n-bit *output*. *S-Box* dengan *input* m-bit dan *output* n-bit dikenal dengan m*n-bit *S-Box*. *S-Box* merupakan bagian yang menjadi pengaman untuk algoritma blok. Semakin besar ukuran *S-Box* yang digunakan maka desain *S-Box* tersebut semakin tahan terhadap serangan kriptoanalisis differensial *(Schneier, 1996)*. Pada *DES*, *S-Box* digunakan untuk memetakan 6-bit *input* menjadi 4-bit *output*. Bit pertama dan keenam digunakan untuk menentukan posisi baris dalam tabel *S-Box*, sedangkan bit kedua sampai kelima digunakan untuk menentukan posisi kolom dalam tabel *S-Box*.

Jaringan Feistel (Feistel Network)

Jaringan *Feistel* dikembangkan pada tahun 1970-an oleh *Horst Feistel*. Jaringan ini banyak digunakan pada algoritma blok. Blok yang panjangnya *n*-bit dibagi menjadi dua bagian, sisi kiri *(L)* dan sisi kanan *(R)*. Masing-masing sisi mempunyai panjang blok *n*/2-bit.

Dalam jaringan *Feistel*, *output* pada *round* ke-*i* ditentukan oleh *output round* sebelumnya. Secara matematik dapat dinyatakan sebagai berikut:

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

K_i merupakan *subkey* yang digunakan pada *round* ke-*i* dan *f* adalah fungsi yang digunakan berulang pada setiap *round*.

Konstruksi jaringan ini memiliki sifat *reversible (dapat dikembalikan)*, sehingga semua algoritma blok yang menggunakan jaringan Feistel tidak perlu mengimplementasikan dua buah algoritma berbeda untuk melakukan proses enkripsi dan dekripsi.

Padding dan *Unpadding*

Proses *padding* adalah penambahan bit-bit isian pada blok terakhir dari *input plaintext* yang akan dienkripsi. *Padding* pada penelitian ini dilakukan dengan cara menambahkan karakter baru yang memiliki kode *ASCII* 1-8 *(Ireland, 2004)*, dan aturannya sebagai berikut:

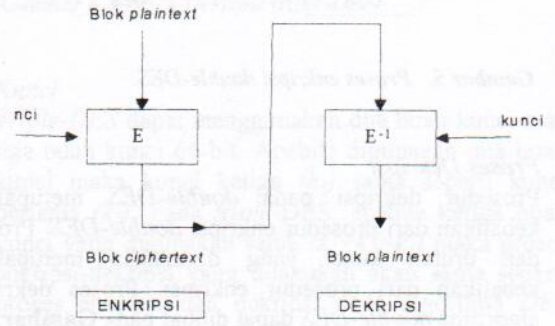
$$n_{Pad} = \left(\left(\left\lfloor \frac{c}{8} \right\rfloor + 1 \right) \times 8 \right) - c$$

dengan *c* adalah jumlah karakter pada blok terakhir dan n_{Pad} adalah jumlah atau kode untuk karakter *padding*. Pada proses dekripsi dilakukan operasi *unpadding*, yaitu

menghilangkan karakter yang ditambahkan saat *padding* pada proses enkripsi.

Electronic CodeBook (ECB)

Electronic CodeBook (ECB) merupakan salah satu modus operasi pada algoritma blok. *ECB* mengenkripsi setiap blok *plaintext* ke *ciphertext*, demikian pula sebaliknya untuk proses dekripsi. Tahapan proses enkripsi dan dekripsi dengan menggunakan modus operasi *ECB* dapat dilihat pada **Gambar 3.**



Gambar 3. *Proses enkripsi-dekripsi dengan modus operasi ECB (Menezes, 1996).*

Kriptoanalisis

Kriptoanalisis *(cryptanalysis)* adalah ilmu untuk mendapatkan *plaintext* dari *ciphertext* tanpa memiliki kunci untuk membuka *ciphertext* tersebut. Orang yang melakukan kriptoanalisis disebut kriptoanalis *(cryptanalysts)*. Sedangkan usaha untuk melakukan kriptoanalisis disebut serangan *(attack)*.

Beberapa jenis serangan yang sering dilakukan oleh kriptoanalis terhadap algoritma blok adalah *(Schneier, 1996)*:

1. *Ciphertext-only attack*.
2. *Known-plaintext attack*.
3. *Chosen-plaintext attack*.

Exhaustive Key Search (Brute-force attack)

Exhaustive key search atau *brute-force attack* adalah suatu teknik dasar yang digunakan kriptoanalis untuk mencoba semua kemungkinan kunci pada sebagian *ciphertext* sampai transformasi dari *ciphertext* menjadi *plaintext* dapat dimengerti dan dapat ditentukan kunci transformasi yang sebenarnya. Pada suatu algoritma blok dengan panjang blok n-bit dan menggunakan kunci k-bit dibutuhkan $\lceil (k+4)/n \rceil$ pasangan *plaintext-ciphertext* yang akan dienkripsi dengan menggunakan kunci k.

Pada kasus terburuk, *exhaustive key search* dapat menemukan kunci k tersebut dengan melakukan dekripsi sebanyak 2^{k-1}.

Kriptoanalisis Differensial dan Kriptoanalisis Linear

Kriptoanalisis Differensial (Differential Cryptanalysis) merupakan metode serangan yang diperkenalkan oleh *Eli Biham* dan *Ali Shamir (1990)*. Metode ini mirip *chosen-plaintext attack*. Cara kerja dari metode ini adalah dengan menganalisa perkembangan dari perbedaan *(difference)* hasil enkripsi pasangan *plaintext* dengan menggunakan kunci yang sama. Sedangkan kriptoanalisis linear *(Linear Cryptanalysis)* merupakan usaha yang dilakukan kriptoanalis untuk menemukan persamaan linear yang efektif sehingga dapat memudahkan dalam menduga transformasi yang dilakukan oleh suatu algoritma.

Analisis Algoritma

Analisis algoritma dilakukan untuk menduga besarnya sumber daya waktu yang dibutuhkan untuk sembarang ukuran *input* n. Dalam penelitian ini, algoritma dievaluasi berdasarkan kompleksitas waktu untuk waktu terburuk, dinotasikan dengan – O *(big O)*.

DESKRIPSI ALGORITMA *TRIPLE*-DES

Algortima *triple-DES* menggunakan algoritma *DES* sebagai algoritma utama. Untuk dapat mengetahui deskripsi algoritma *triple-DES*, terlebih dahulu diberikan deskripsi algoritma *DES* dan *double-DES (sebagai varian multiple-DES yang lebih sederhana dibanding triple-DES)*.

Data Encryption Standard (DES)

Algoritma *DES* terdiri atas dua bagian utama yaitu ekspansi kunci *(key expansion)* dan proses enkripsi-dekripsi.

1. Ekpansi kunci

Tahapan proses pembentukan subkunci adalah:

a. Kunci 64-bit dipermutasikan pada tabel PC *(permuted choice)* 1 untuk menghasilkan 56-bit kunci.
b. *Output* dari PC-1 kemudian dibagi dua menjadi C_0 dan D_0 (masing-masing 28-bit). Setelah C_0 dan D_0 terdefinisi, dibuat blok C_n dan D_n, $1 \le n \le 16$. Setiap pasangan C_n dan D_n dibentuk dari pasangan C_{n-1} dan D_{n-1} sebelumnya secara

berurutan dengan menggunakan penjadwalan *left shift* dibawah ini:
- Untuk *round* 1, 2, 9, dan 16, C_n dan D_n digeser 1-bit ke kiri
- Selain *round* diatas, C_n dan D_n digeser 2-bit ke kiri.

c. Setelah selesai, C_n dan D_n setiap iterasi digabung *(concatenation)* menjadi C_nD_n.

d. Kemudian untuk membentuk subkunci-subkunci *(k_n untuk $1 \le n \le 16$)*, permutasikan 56-bit C_nD_n pada **tabel PC-2** untuk menghasilkan 48-bit k_n.

2. *Proses Enkripsi*

Input proses enkripsi merupakan *plaintext* 64-bit, misalkan X. Untuk mengenkripsi X dilakukan langkah-langkah sebagai berikut:

a. *Input* X terlebih dahulu dipermutasikan dengan menggunakan tabel IP *(initial permutation)*.

b. *Output* dari tabel IP kemudian dibagi menjadi dua bagian yaitu L_0 dan R_0 *(masing-masing 32-bit)*.

c. Untuk $n = 1...16$
Tentukan L_n dan R_n dengan menggunakan persamaan berikut ini:

$$L_n = R_{n-1}$$
$$R_n = L_{n-1} \oplus f(R_{n-1}, k_n)$$

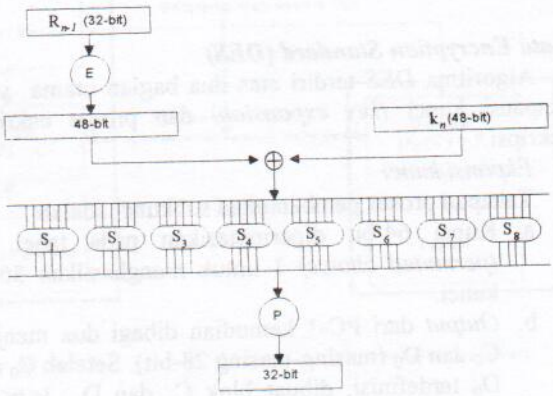
Tahapan proses untuk menentukan fungsi f dapat dilihat pada **Gambar 4.**

d. Pada akhir iterasi *($n=16$)* diperoleh blok L_{16} dan R_{16} *(masing-masing 32-bit)*. Kedua blok tersebut dibalik urutannya *(reverse)* menjadi $R_{16}L_{16}$.

e. Untuk memperoleh *output* X *(ciphertext)*, $R_{16}L_{16}$ dipermutasikan dengan menggunakan **tabel IP^{-1}.**

3. *Proses dekripsi*

Proses dekripsi *DES* sama seperti proses enkripsinya, hanya membalik urutan subkunci yang digunakan.

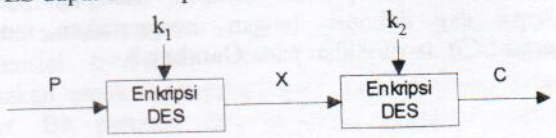


Gambar 4. Proses perhitungan fungsi f.

Double-DES

1. *Proses Enkripsi*

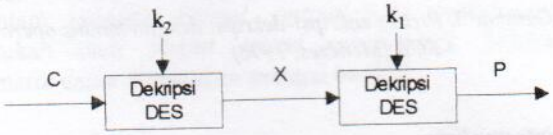
Pada *double-DES*, prosedur enkripsi mirip proses enkripsi pada *DES* hanya saja prosesnya diulang sebanyak dua kali. Proses enkripsi algoritma *double-DES* dapat dilihat pada **Gambar 5.**



Gambar 5. Proses enkripsi double-DES.

2. *Proses Dekripsi*

Prosedur dekripsi pada *double-DES* merupakan kebalikan dari prosedur enkripsi *double-DES*. Proses dan urutan kunci yang digunakan merupakan kebalikan dari prosedur enkripsi. Proses dekripsi algoritma *double-DES* dapat dilihat pada **Gambar 6.**



Gambar 6. Proses dekripsi double-DES.

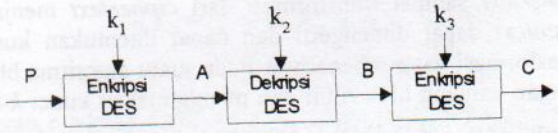
3. *Kunci*

Double-DES menggunakan dua buah kunci 64-bit untuk proses enkripsi dan dekripsinya. Pada setiap tahap, kunci 64-bit digunakan untuk membangkitkan 16 subkunci yang akan digunakan pada setiap *round DES*.

Triple-DES

1. *Proses Enkripsi*

Pada *triple-DES*, prosedur enkripsi mirip proses enkripsi pada *DES* hanya saja prosesnya diulang sebanyak tiga kali. Proses enkripsi algoritma *triple-DES* dapat dilihat pada **Gambar 7.**



Gambar 7. Proses Enkripsi triple-DES

2. *Proses Dekripsi*
   Pada *triple-DES*, prosedur dekripsi merupakan kebalikan dari prosedur enkripsi *triple-DES*. Proses dan urutan kunci yang digunakan merupakan kebalikan dari prosedur enkripsi. Proses dekripsi algoritma *triple-DES* dapat dilihat pada **Gambar 8.**

Gambar 8. *Proses Dekripsi triple-DES*

3. *Kunci*
   Triple-DES dapat menggunakan dua buah kunci atau tiga buah kunci 64-bit. Apabila digunakan dua buah kunci maka kunci ketiga (k_3) sama seperti kunci pertama (k_1). Pada *triple*-DES, apabila ketiga buah kunci yang digunakan sama ($k_1=k_2=k_3$) maka proses enkripsi-dekripsi yang dilakukan akan sama seperti proses enkripsi dan dekripsi pada algoritma *DES* biasa.

HASIL DAN PEMBAHASAN

Triple-DES merupakan varian *DES* yang dikembangkan untuk mengatasi kelemahan yang terdapat pada *DES*. Oleh sebab itu, untuk dapat mengetahui kinerja dari algoritma *triple-DES* juga dilakukan analisis terhadap algoritma *DES* dan *double-DES*. Pada sub-bab berikut akan dijelaskan analisis yang telah dilakukan meliputi analisis algoritma, analisis keamanan dan analisis hasil implementasi.

A. Analisis Algoritma

Analisis algoritma dilakukan dengan asumsi bahwa mesin yang digunakan adalah model *Random-Access Machine (RAM)*, berprosesor tunggal dan lamanya waktu eksekusi setiap operasi tersebut adalah satu satuan waktu. Pada mesin ini, instruksi-instruksi program dieksekusi baris demi baris secara berurutan *(Cormen et al., 1990).*

A.1. Analisis Algoritma Enkripsi

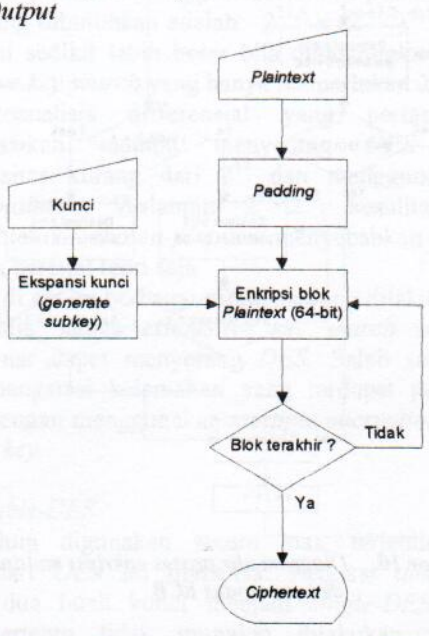
Double-DES dan *triple-DES* merupakan varian *DES* dengan menggunakan *multiple-DES (m-DES)*, sehingga analisis untuk *double-DES* dan *triple-DES* digeneralisasi menjadi analisis *multiple-DES*. Diagram alir proses enkripsi *DES* dengan *ECB* dapat dilihat pada **Gambar 9,** sedangkan diagram alir proses enkripsi untuk *multiple-DES (double-DES* dan *triple-DES)* dapat dilihat pada **Gambar 10.**

1. Langkah untuk melakukan proses enkripsi modus *ECB* pada *DES* adalah:
   a. *Padding plaintext*
   b. Ekspansi kunci
   c. Enkripsi blok *plaintext* 64-bit
   d. *Output*

   Waktu eksekusi pada langkah a, b, dan d adalah konstan *(misalkan α)* karena tidak dipengaruhi ukuran *input*. Sedangkan langkah c dipengaruhi jumlah blok *input plaintext* 64-bit, sehingga untuk *input* berukuran *n*-blok diperlukan waktu εn. Jadi, notasi-*O* untuk kasus terburuk proses enkripsi *DES* dengan modus *ECB* adalah:

$$E_{(ECB\_DES)} = \varepsilon n + \alpha \in O(n)$$

2. Langkah untuk melakukan proses enkripsi modus *ECB* pada *multiple-DES (khususnya double-DES* dan *triple-DES)* adalah:
   a. *Padding plaintext*
   b. Untuk $m = 2, 3, ..., p$
      b.1. Ekspansi kunci
      b.2. Enkripsi blok *plaintext* 64-bit
   c. *Output*

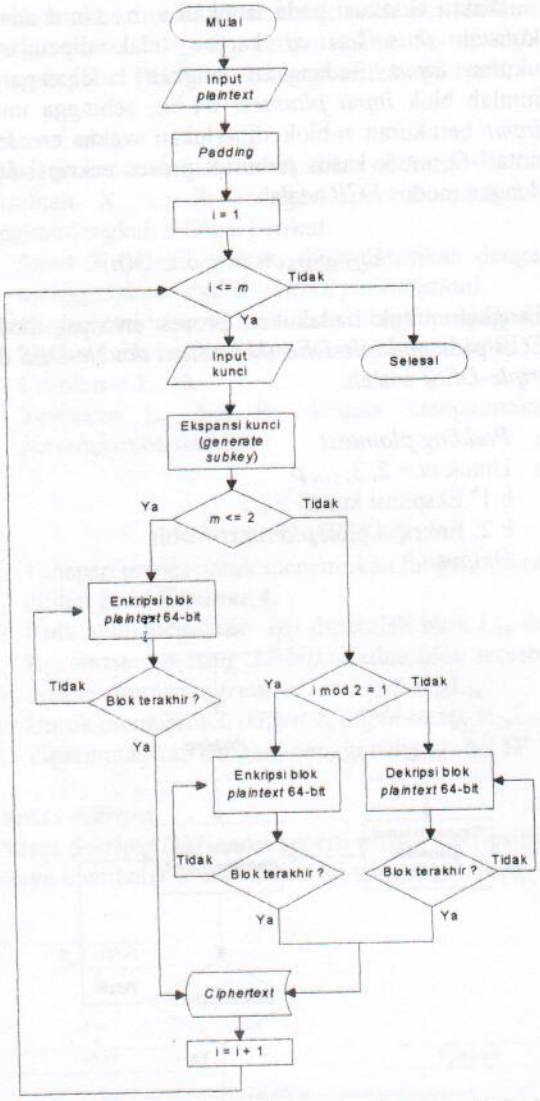


Gambar 9. *Diagram alir proses enkripsi DES dengan modus ECB.*

Pada percobaan ini dilakukan untuk m=2 *(double-DES)* dan m=3 *(triple-DES)*. Waktu eksekusi pada langkah a, b1, dan c adalah konstan *(misalkan α)* karena tidak dipengaruhi ukuran input. Sedangkan

langkah b2 dipengaruhi oleh jumlah blok input plaintext 64-bit dan jumlah ulangan *DES (m-DES)*, sehingga untuk input berukuran n-blok diperlukan waktu m(*εn*). Jadi, *notasi-O* untuk kasus terburuk proses enkripsi *multiple-DES* dengan modus *ECB* adalah:

$$E_{(ECB\_multiple\text{-}DES)} = m(\varepsilon n + \alpha) = m\ O(n) \in O(n)$$



Gambar 10. *Diagram alir proses enkripsi multiple-DES dengan modus ECB.*

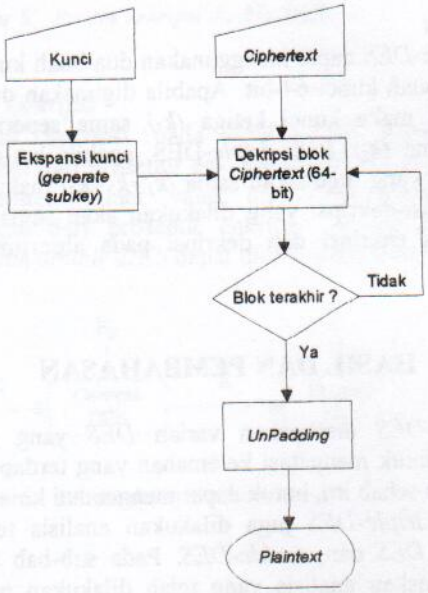
A.2. Analisis Algoritma Dekripsi

Diagram alir proses dekripsi *DES* dengan modus *ECB* dapat dilihat pada **Gambar 11**, sedangkan diagram alir untuk proses dekripsi *multiple-DES* (*double-DES* dan *triple-DES)* dapat dilihat pada **Gambar 12**.

1. Langkah untuk melakukan proses dekripsi modus *ECB* pada *DES* adalah:
   a. Ekspasi kunci
   b. Dekripsi blok *ciphertext* 64-bit
   c. *Unpadding*
   d. *Output*

   Waktu eksekusi pada langkah a, c, dan d adalah konstan *(misalkan β)* karena tidak dipengaruhi ukuran *input*. Sedangkan langkah b dipengaruhi jumlah blok *input ciphertext* 64-bit, sehingga untuk *input* berukuran *n*-blok diperlukan waktu δ*n*. Jadi, *notasi-O* untuk kasus terburuk proses dekripsi *DES* dengan modus *ECB* adalah:

$$D_{(ECB\_DES)} = \delta n + \beta \in O(n)$$

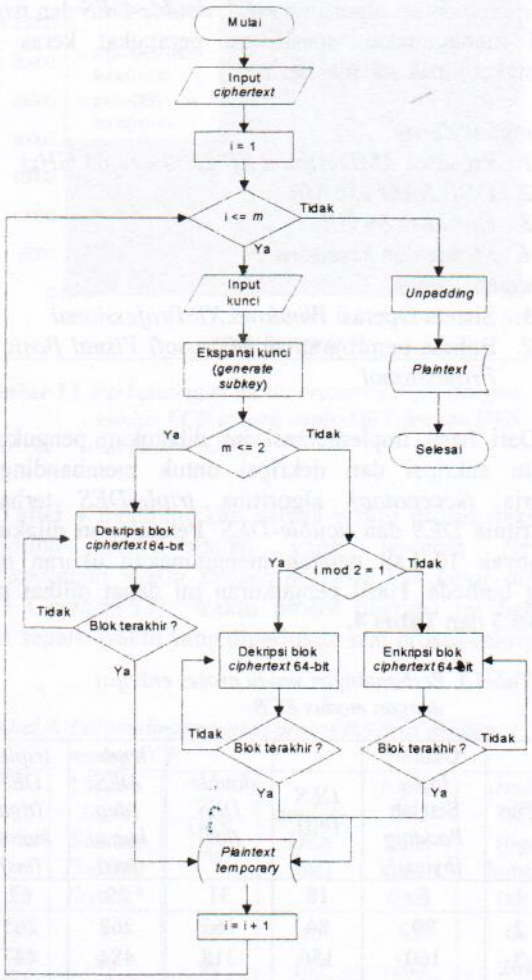


Gambar 11. *Diagram alir proses dekripsi DES dengan modus ECB.*

2. Langkah untuk melakukan proses dekripsi modus *ECB* pada *multiple-DES (khususnya double-DES dan triple-DES)* adalah:
   a. Untuk $m = 2, 3, ..., p$
      a.1. Ekspansi kunci
      a.2. Dekripsi blok *plaintext* 64-bit
   b. *Unpadding*
   c. *Output*

   Pada percobaan ini dilakukan untuk m=2 (*double-DES)* dan m=3 (*triple-DES)*. Waktu eksekusi pada langkah a1, b, dan c adalah konstan *(misalkan β)* karena tidak dipengaruhi ukuran *input*. Sedangkan langkah a2 dipengaruhi oleh jumlah blok *input ciphertext* 64-bit dan jumlah ulangan *DES* (*m-DES)*, sehingga untuk *input* berukuran *n*-blok diperlukan waktu *m*(δ*n*). Jadi, notasi-O untuk kasus terburuk proses dekripsi *multiple-DES* dengan modus *ECB* adalah:

$$D_{(ECB\_multiple\text{-}DES)} = m(\delta n + \beta) = m\, O(n) \in O(n)$$



Gambar 12. *Diagram alir proses dekripsi multiple-DES dengan modus ECB*

B. *Analisis Keamanan*

B.1. DES (Data Encryption Standard)

Beberapa serangan yang dilakukan kriptoanalis terhadap *DES*, hampir semuanya memanfaatkan kelemahan kunci yang digunakan pada *DES*. Selain ukuran kunci, nilai *S-Box* yang konstan juga dimanfaatkan oleh kriptoanalis untuk melakukan serangan, tetapi sampai saat ini belum ada publikasi kelemahan yang dianggap fatal pada *S-Box (Stallings, 2003)*.

Jenis serangan yang paling mudah dilakukan oleh kriptoanalis terhadap *DES* adalah *exhaustive key search*. Untuk dapat melakukan serangan ini kriptoanalis memerlukan minimal $\lceil (56+4)/64 \rceil = 1$ pasangan *plaintext-ciphertext*. Pada kasus terburuk, dibutuhkan sebanyak $2^{56-1}=2^{55}$ proses dekripsi untuk mendapatkan kunci yang digunakan pada proses enkripsi.

Karena *exhaustive key search* diangap kurang efisien dalam menyerang algoritma *DES*, kriptoanalis mulai mengembangkan jenis *attack* yang lain, yaitu kriptoanalisis linear dan kriptoanalisis differensial. Dari kedua jenis kriptoanalisis tersebut, hanya kriptoanalisis differensial yang diangap mampu menyerang *DES* dengan kompleksitas yang lebih baik dari *exhaustive key search*.

Dari penelitian *Biham & Shamir (1991)* tentang kriptoanalisis differensial pada 16-*round DES* disimpulkan bahwa:

Pada *DES* digunakan jaringan Feistel dengan 16-*round*. Ini berarti usaha yang diperlukan oleh kriptoanalis untuk dapat mematahkan *DES* dengan menggunakan kriptoanalisis differensial adalah:

- *chosen-plaintext* = $2^{47,2}$
- *plaintext* yang dianalisa = $2^{47,2} \times 2^{-10,72} = 2^{36,45}$
- kompleksitas analisa = $2^{47,2} \times 2^{-10} = 2^{37,2}$

Apabila kriptoanalis menggunakan *known-plaintext* maka usaha yang dibutuhkan adalah: $2^{31,5} \times (2^{-47,2})^{-0,5} = 2^{55,1}$. Usaha ini sedikit lebih besar bila dibandingkan dengan *exhaustive key search* yang hanya memerlukan 2^{55}.

Kriptoanalisis differensial yang pertama kali dipublikasikan mampu menyerang *DES* dengan kompleksitas kurang dari 2^{55} dan menggunakan 2^{47} *chosen-plaintext*. Walaupun $2^{47}<2^{55}$, kesulitan untuk memperoleh 2^{47} *chosen-plaintext* menyebabkan serangan ini hanya bersifat teori saja.

Jadi, di antara berbagai serangan yang dilakukan oleh kriptoanalis, hanya *exhaustive key search* saja yang benar-benar dapat menyerang *DES*. Salah satu solusi untuk mengatasi kelemahan yang terdapat pada *DES* adalah dengan menggunakan *multiple encryption* dengan *multiple key*.

B.2. double-DES

Sebelum digunakan secara luas, terlebih dahulu varian dari *DES* ini dianalisa. Reduksi *double-DES* dengan dua buah kunci menjadi *single-DES* dengan kunci tertentu tidak mungkin dilakukan. Hal itu disebabkan oleh sifat *DES* yang tidak tertutup (*not closed*). Pembuktian *DES* bersifat tidak tertutup *(Campbell dan Wiener, 1992)* dimulai dengan membuktikan bahwa *DES* bersifat tertutup kemudian dicari kontradiksinya.

Walaupun penggunaan *double-DES* tidak dapat direduksi menjadi *single-DES*, *double-DES* belum bisa

memberikan keamanan seperti yang diharapkan. Kunci efektif 112-bit yang diharapkan dapat mengatasi kelemahan pada DES kurang tahan dalam mengamankan informasi. Kriptoanalis dengan menggunakan *meet-in-the-middle attack* berhasil menyerang teknik kriptografi ini dengan usaha 2^{56} Terlebih lagi, apabila *meet-in-the-middle attack* yang dilakukan pada 2 pasangan *plaintext-ciphertext* mengakibatkan kemungkinan (*probability*) kunci yang benar adalah $1\text{-}2^{-16}$=0,999. Hasil ini menunjukkan bahwa *known-plaintext attack* pada *meet-in-the-middle attack* berhasil menyerang *double*-DES.

B.3. triple-DES

Usaha nyata untuk mengatasi kelemahan pada *double*-DES terhadap *meet-in-the-middle attack* adalah dengan menggunakan tiga tahap enkripsi. Ini akan meningkatkan usaha *known-plaintext attack* dan kriptoanalisis differensial. Berdasarkan hasil penelitian *Coppersmith (1994)*, usaha yang dibutuhkan oleh *brute-force key search* pada *triple-DES* dengan menggunakan dua buah kunci adalah 2^{112} dan diduga bahwa usaha untuk kriptoanalisis differensial meningkat melebihi 10^{52} (*chosen-plaintext*). Pada keadaan yang demikian, ukuran dari ruang pesan menjadi faktor pembatas keamanan. Menurut *Stallings (2003)*, sampai saat ini belum ada kriptoanalisis yang mampu menyerang *triple*-DES.

Berdasarkan **Tabel 2,** *exhaustive key search* yang dilakukan kriptoanalis terhadap *triple-DES* dengan dua kunci *(112-bit)* memerlukan waktu yang hampir tidak mungkin dilakukan oleh kriptoanalis. Walaupun demikian, beberapa peneliti menyarankan penggunaan *triple-DES* dengan tiga buah kunci untuk lebih meningkatkan keamanan yang dihasilkan.

***Tabel 2.** Waktu rata-rata yang dibutuhkan untuk exhaustive key search (Stallings, 2003)*

Key Size (bits)	*Number of alternatif key*	*Time required at 1 encryption/μs*	*Time required at 10^6 encryption/μs*
32	$2^{32} = 4{,}3\text{x}10^{9}$	2^{31} μs = 35,8 menit	2,15 ms
56	$2^{56} = 7{,}2\text{x}10^{16}$	2^{55} μs = 1142 thn	10,01 thn
112	$2^{112} = 5{,}2\text{x}10^{33}$	2^{111} μs = $8{,}2\text{x}10^{19}$ thn	$8{,}2\text{x}10^{13}$ thn
128	$2^{128} = 3{,}4\text{x}10^{38}$	2^{127} μs = $5{,}4\text{x}10^{24}$ thn	$5{,}4\text{x}10^{18}$ thn
168	$2^{168} = 3{,}7\text{x}10^{50}$	2^{167} μs = $5{,}9\text{x}10^{36}$ thn	$5{,}9\text{x}10^{30}$ thn

C. *Analisis Hasil Implementasi*

Implementasi algoritma *DES*, *double-DES* dan *triple-DES* menggunakan spesifikasi perangkat keras dan perangkat lunak sebagai berikut:

Perangkat Keras

1. Prosesor *AMD Athlon XP 2000+ (1,63 GHz)*
2. *DDR RAM 256 MB*
3. *Harddisk 80 GB*
4. *Mouse* dan *keyboard*

Perangkat Lunak

1. Sistem Operasi *Windows XP Professional*
2. Bahasa pemrograman *Microsoft Visual Basic 6.0 Professional*

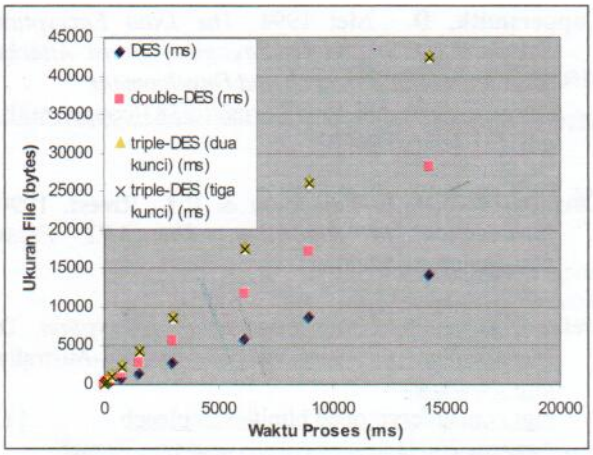
Dari hasil implementasi ini dilakukan pengukuran waktu enkripsi dan dekripsi untuk membandingkan kinerja *(kecepatan)* algoritma *triple-DES* terhadap algoritma *DES* dan *double-DES*. Pengukuran dilakukan sebanyak 10 kali dengan menggunakan ukuran *input* yang berbeda. Hasil pengukuran ini dapat dilihat pada **Tabel 3** dan **Tabel 4.**

***Tabel 3.** Perbandingan waktu proses enkripsi dengan modus ECB*

File	Ukuran *Input* Setelah *Padding (bytes)**	*DES (ms)*	*double-DES (ms)*	*triple-DES (dua kunci) (ms)*	*triple-DES (tiga kunci) (ms)*
1	8	18	31	59	62
2	80	84	165	262	265
3	160	156	318	484	487
4	344	331	662	1006	1003
5	760	718	1447	2159	2159
6	1512	1421	2849	4284	4287
7	3008	2837	5694	8506	8525
8	6144	5878	11734	17540	17543
9	9008	8731	17396	26015	26043
10	14296	14174	28359	42306	42312

** Diasumsikan file input dan output untuk proses enkripsi telah mengalami proses padding dan dikonversi ke heksadesimal.*

Tabel 3 menunjukkan bahwa waktu proses enkripsi algoritma *DES* adalah yang tercepat *(grafik perbandingan waktu proses enkripsi ini dapat dilihat pada Gambar 13)*. Antara *triple-DES* dengan dua kunci dan *triple-DES* dengan tiga kunci tidak terdapat perbedaan waktu proses enkripsi yang signifikan, tetapi *triple-DES* dengan tiga kunci *(168-bit)* mampu memberikan perlindungan informasi yang lebih baik dibanding *triple-DES* dengan dua kunci *(112-bit)*.



***Gambar 13**. Perbandingan waktu proses enkripsi dengan modus ECB antara triple-DES dengan DES dan double-DES.*

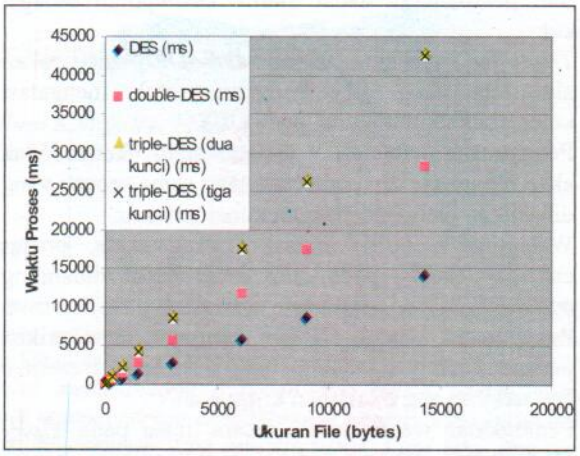
Tabel 4 menunjukkan bahwa waktu proses dekripsi algoritma *DES* adalah yang tercepat *(grafik perbandingan waktu proses dekripsi ini dapat dilihat pada Gambar 14)*. Waktu proses dekripsi ini hampir sama seperti waktu yang diperlukan saat proses enkripsi.

***Tabel 4**. Perbandingan waktu proses dekripsi dengan modus ECB*

File	Ukuran *Input* Setelah *Padding (bytes)**	DES (ms)	*double*-DES (ms)	*triple*-DES (dua kunci) (ms)	*triple*-DES (tiga kunci) (ms)
1	8	28	46	62	62
2	80	99	187	265	268
3	160	171	328	490	490
4	344	337	681	1009	1006
5	760	731	1450	2168	2165
6	1512	1440	2859	4296	4287
7	3008	2853	5703	8503	8522
8	6144	5896	11724	17550	17559
9	9008	8759	17412	26178	26174
10	14296	14303	28384	42497	42462

** Diasumsikan file input dan output untuk proses dekripsi telah mengalami proses padding dan dikonversi ke heksadesimal.*

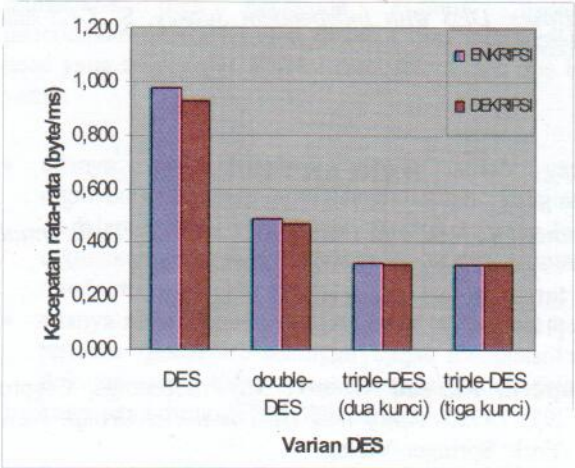
Tabel 5 menunjukkan bahwa kecepatan rata-rata proses enkripsi dan dekripsi menurun apabila pengulangan *DES* semakin besar *(nilai m pada multiple-DES)*. Sedangkan kecepatan rata-rata antara proses enkripsi dan dekripsi pada masing- masing algoritma hampir bisa dikatakan sama. Grafik yang menunjukkan perbandingan kecepatan rata-rata antara *DES*, *double-DES* dan *triple-DES* dapat dilihat pada **Gambar 15.**



***Gambar 14**. Perbandingan waktu proses dekripsi dengan modus ECB antara triple-DES dengan DES dan double-DES.*

***Tabel 5**. Perbandingan kecepatan rata-rata proses enkripsi dan dekripsi*

Algoritma	Kecepatan rata-rata (byte / ms)	
	Enkripsi	Dekripsi
DES	0,973	0,926
double-DES	0,490	0,472
triple-DES (dua kunci)	0,321	0,319
triple-DES (tiga kunci)	0,319	0,318



***Gambar 15**. Perbandingan kecepatan rata-rata proses enkripsi-dekripsi antara triple-DES dengan DES dan double-DES.*

KESIMPULAN DAN SARAN

Kesimpulan

Dari pembahasan dapat ditarik kesimpulan sebagai berikut:

1. *Triple-DES* sebagai varian dari *DES* merupakan algoritma yang dikembangkan untuk mengatasi kelemahan ukuran kunci pada *DES*.
2. Penggunaan *triple-DES* dengan suatu kunci tidak akan menghasilkan pemetaan yang sama seperti yang dihasilkan oleh *DES* dengan kunci tertentu.
3. Walaupun memiliki notasi-*O* yang sama, proses enkripsi-dekripsi pada *DES* lebih cepat dibanding *triple-DES*.
4. Penggunaan *triple-DES* mampu memberikan perlindungan yang cukup baik terhadap *exhaustive key search* yang dilakukan kriptoanalis.
5. Peningkatan waktu proses secara linear pada *triple-DES* diikuti dengan peningkatan keamanan secara eksponensial.

Saran

Berikut adalah beberapa saran untuk penelitian lebih lanjut:

1. Penelitian selanjutnya dapat dikembangkan dengan menggunakan modus operasi yang lain seperti *Cipher Block Chainning*, *Cipher Feedback* dan *Output Feedback*.
2. Pada penelitian ini hanya difokuskan pada *exhaustive key search*. Untuk penelitian selanjutnya dapat dianalisa lebih lanjut kinerja *triple-DES* terhadap kriptoanalisis yang lain *(misalnya kriptoanalisis linear dan differensial)*.
3. Untuk penelitian selanjutnya dapat dianalisa varian *DES* yang lain seperti *DES with Key-Dependent S-Boxes*, *DES with Independent Subkey*, S^n *DES* dan lain-lain.

DAFTAR PUSTAKA

Biham, E. & A. Shamir. 1991. *Differential Cryptanalysis of The Full 16-round* DES. http://citeseer.ist.psu.edu/123842.html [12 Agustus 2004].

Campbell, K., and Wiener, M. Proceedings, Crypto '92, 1992. *Proof that DES is not a Group.* New York: Springer-Verlag. www3.sympatico.ca/wienerfamily/Michael/MichaelPapers/desgroup.pdf [3 Februari 2004].

Coppersmith, D. Mei 1994. *The Data Encryption Standard (DES) and Its Strength Against Attacks. IBM Journal of Research and Development.* www.research.ibm.com/journal/rd/383/coppersmith.pdf [31 Januari 2003].

Cormen, T.H., C.E. Leiserson & R.L. Rivest. 1990. *Introduction to Algorithm. The MIT Press*, Massachusetts-London.

Ireland, D. 2004. *Using Padding in Encryption.* DI Management Services. Sydney-Australia. http://www.di-mgt.com.au/cryptopad.html#exampleecb [12 Agustus 2004].

Keller, S. S. 2000. *Modes of Operation Validation System for The Triple Data Encryption Algorithm (TMOVS): Requirements and Procedures. NIST* Special Publication 800-20.

http://csrc.nist.gov/publications/nistpubs/800-20/800-20.pdf [11Desember 2003].

Mardiana, D. 2002. *Algoritma Blowfish untuk Penyandian Pesan.* Skripsi. Jurusan *Ilmu Komputer IPB.*

Menezes, A. J., P. V. Oorschot and S. Vanstone. 1996. *Handbook of Applied Cryptography.* CRC Press Inc.

Oorschot, P., and Wiener, M. Proceedings, Eurocrypt '90, 1990. *A Known-Plaintext Attack on Two-Key Triple Encryption.* New York: Springer-Verlag . www3.sympatico.ca/wienerfamily/Michael/MichaelPapers/TwokeytripleDES.pdf [3 Februari 2004].

Schneier, B. 1996. *Applied Cryptography Second Edition: Protocols, Algorithms and Source Code in C.* New York: Wiley.

Stallings, W. 2003. *Cryptography and Network Security: Princliples and Practice 3rd edition.* Prentice-Hall Inc. New Jersey.

_____. DES *Encryption.* http://www.tropsoft.com/strongenc/des.htm [11 Desember 2003].

_____. *Triple* DES *Encryption.* http://www.tropsoft.com/strongenc/des3.htm [11 Desember 2003].