DIFFERENSIAL CRYPTANALISIS PADA SDES dan DES 8 PUTARAN

SKRIPSI

Sebagai salah satu syarat untuk memperoleh gelar
Sarjana dalam bidang Ilmu Komputer

oleh:
ANNORA DIONE CHRISTABEL
0510960009-96

PROGRAM STUDI ILMU KOMPUTER
JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS BRAWIJAYA
MALANG

2010


UNIVERSITAS BRAWIJAYA

LEMBAR PENGESAHAN SKRIPSI

DIFFERENSIAL CRYPTANALISIS PADA SDES dan DES 8 PUTARAN

Oleh:
ANNORA DIONE CHRISTABEL
0510960009-96

Setelah dipertahankan di depan Majelis Penguji
Pada tanggal 21 Januari 2010
dan dinyatakan memenuhi syarat untuk memperoleh gelar
Sarjana dalam bidang Ilmu Komputer

Pembimbing 1,	Pembimbing 2,
Bayu Rahayudi, ST., MT	**Reza Andria Siregar, ST**
NIP. 197407122006041001	**NIP. 197906212006041003**

Mengetahui,
Ketua Jurusan Matematika
Fakultas MIPA Universitas
Brawijaya

Dr. Agus Suryanto, MSc.
NIP. 196908071994121001


UNIVERSITAS BRAWIJAYA

LEMBAR PERNYATAAN

Saya yang bertanda tangan di bawah ini :

Nama : ANNORA DIONE CHRISTABEL

NIM : 0510960009-96

Jurusan : Matematika

Penulis skripsi berjudul :DIFFERENSIAL CRYPTANALISIS PADA SDES dan DES 8 PUTARAN

Dengan ini menyatakan bahwa :

1. **Isi dari skripsi yang saya buat adalah benar-benar karya sendiri dan tidak menjiplak karya orang lain, selain nama-nama yang termaktub di isi dan tertulis di daftar pustaka dalam skripsi ini.**

2. **Apabila dikemudian hari ternyata skripsi yang saya tulis terbukti hasil jiplakan, maka saya akan bersedia menanggung segala resiko yang akan saya terima.**

Demikian pernyataan ini dibuat dengan segala kesadaran.

Malang, 21 Januari 2010
Yang menyatakan,

(Annora Dione C)
NIM. 0510960009

DIFFERENSIAL CRYPTANALISIS PADA SDES dan DES 8 PUTARAN

ABSTRAK

Keamanan informasi sangat penting, untuk menjaga keamanan data atau informasi salah satunya adalah dengan menyandikan data atau informasi tersebut sehingga tidak dapat dibaca oleh orang yang tidak berhak mengetahui isi data tersebut. Kriptografi adalah ilmu dan seni untuk menjaga keamanan pesan. Dimana pesan yang dapat dibaca (*plaintext*) diubah menjadi pesan yang disandikan sehingga tidak memiliki makna lagi (*ciphertext*). Salah satu algoritma kriptografi adalah DES. *Data Encryption Standart* (DES) merupakan perkembangan awal kriptografi pada bidang komputer. Selain kriptografi terdapat algoritma yang berlawanan dengan kriptografi yaitu kriptanalisis dimana tujuan dari algoritma ini adalah untuk mendapatkan kunci atau juga untuk mendapatkan informasi yang telah disandikan dengan algoritma kriptografi. Kriptanalisis dapat juga digunakan untuk menguji ketahanan suatu algoritma kriptografi. Salah satu algoritma kriptanalisis adalah *differensial cryptanalysis. Differensial cryptanalysis* adalah usaha untuk menemukan *subkey* putaran terakhir dimana *differensial cryptanalysis* termasuk jenis *choses plaintext attack.* Didalam skripsi ini digunakan algoritma kriptografi yaitu SDES dan DES 8 putaran serta algoritma kriptanalisis yaitu *differensial cryptanalysis*. Tujuan skripsi ini adalah untuk mendapatkan *subkey* putaran terakhir pada SDES dan DES 8 putaran menggunakan *differensial cryptanalysis.* Dimana akan dihasilkan *subkey* sebesar 8 bit dari 10 bit kunci asli pada SDES dan 48 bit dari 64 bit kunci asli pada DES 8 putaran.


UNIVERSITAS BRAWIJAYA

DIFFERENSIAL CRYPTANALISIS IN SDES AND DES 8 ROUND

ABSTRACT

Information security is an important matter. One method to keep data or information secure is to encode it. Thus, it cannot be read by other people who do not have access to know the data or information. Cryptography is a science and art to keep messages secure. A readable message (plaintext) is changed to an encoded one so it seems not to have meaning (cipher text).

One form of cryptographic algorithm is Data Encryption Standard (DES). DES is the initial development of cryptography in computer. Other than cryptography, there is an adversative algorithm named cryptanalysis. The aim of cryptanalysis is to get the key or information that has been previously encoded with cryptographic algorithm. Cryptanalysis can also be employed to test a cryptographic algorithm's tenacity.

One form of cryptanalysis algorithm is differential cryptanalysis. It is a mean to find a sub key in the final lap where differential cryptanalysis is the chosen plaintext attack. This thesis incorporates SDES and 8-laps DES cryptographic algorithm and differential cryptanalysis. The purpose of this thesis is to obtain the last lap's sub key of SDES and 8-laps DES using differential cryptanalysis. The results given are 8-bit key of 10-bit original key in SDES and 48-bit key of 64-bit original key in 8-laps DES.

Kata Pengantar

Puji syukur penulis panjatkan kehadirat Tuhan YME, yang telah memberikan rahmat dan berkat-Nya, sehingga skripsi yang berjudul " DIFFERENSIAL CRYPTANALISIS PADA SDES dan DES 8 putaran " ini dapat berjalan dengan baik. Skripsi ini disusun dan diajukan sebagai syarat untuk memperoleh gelar sarjana pada program studi Ilmu Komputer, jurusan Matematika, fakultas MIPA, universitas Brawijaya.

Dalam penyelesaian Skripsi ini, penulis telah mendapat begitu banyak bantuan baik moral maupun materiil dari banyak pihak. Atas bantuan yang telah diberikan, penulis ingin menyampaikan penghargaan dan ucapan terima kasih yang sedalam-dalamnya kepada:

1. Bayu Rahayudi, ST., MT dan Reza Andria S, ST selaku pembimbing. Terima kasih atas semua saran, bantuan, waktu dan bimbingannya.
2. Drs. Marji,MT selaku Ketua Program Studi Ilmu Komputer.
3. Kasyful Amron, ST selaku Penasihat Akademik.
4. Dr. Agus Suryanto, MSc selaku Ketua Jurusan Matematika.
5. Segenap bapak dan ibu dosen yang telah memberikan ilmunya kepada penulis.
6. Segenap staf dan karyawan di Jurusan Matematika FMIPA Universitas Brawijaya.
7. Bapak, Ibu, kakak dan adik. Terima kasih atas doa, dukungan dan semangat yang tiada henti.
8. Teman-teman yang sudah banyak membantu terselesaikannya skripsi ini.
9. Pihak lain yang telah membantu terselesaikannnya skripsi ini yang tidak bisa penulis sebutkan satu-persatu.

Semoga penulisan laporan skripsi ini bermanfaat bagi pembaca sekalian. Penulis menyadari bahwa skripsi ini masih jauh dari kesempurnaan, dan mengandung banyak kekurangan,

sehingga dengan segala kerendahan hati penulis mengharapkan kritik dan saran yang membangun dari pembaca.

Malang, 21 Januari 2010

Penulis

DAFTAR ISI

Halaman

Daftar Gambar

Daftar Tabel

Daftar Sourcecode

Daftar Lampiran

BAB I
PENDAHULUAN

1.1 Latar Belakang

Ilmu pengetahuan dan teknologi telah berkembang dengan pesatnya. Dengan adanya perkembangan teknologi ini sangat mempengaruhi sistem informasi yang ada. Informasi yang ada dapat dengan mudah dikirimkan dari satu orang ke orang yang lain. Oleh karena itu keamanan informasi sangat penting untuk menjaga keamanan informasi dari orang-orang yang tidak berhak mengetahuinya. Salah satu cara menjaga keamanan data atau informasi adalah dengan menyandikan data atau informasi tersebut sehingga tidak dapat dibaca oleh orang yang tidak berhak mengetahui isi data tersebut.

Kriptografi adalah ilmu dan seni untuk menjaga keamanan pesan. Proses penyandian pesan yang dapat dibaca (*plaintext*) menjadi pesan yang disandikan sehingga tidak memiliki makna lagi (*ciphertext*) disebut enkripsi. Sedangkan proses mengembalikan *ciphertext* menjadi *plaintext* disebut dekripsi. Tujuan utama kriptografi adalah keamanan dimana pengirim informasi atau data dapat memastikan bahwa hanya penerima yang dituju yang dapat mendekripsikan data atau informasi yang dienkripsi oleh pengirim *(www.dwiantoro.com)*.

Data Encryption Standart (DES) merupakan perkembangan awal kriptografi pada bidang komputer dimana mekanisme kriptografi ini kemudian menjadi suatu standar keamanan data (*www.smeapgri-tng.sch.id*). DES mengenkripsi *plaintext* sebesar 64 bit (8 byte) dengan panjang kunci sekitar 56 bit (7 byte) sebanyak 16 putaran (Budi,2004).

SDES adalah versi yang lebih rendah dari DES. SDES memiliki bagian-bagian yang sama seperti DES tetapi hanya dengan blok dan ukuran kunci yang lebih kecil (dengan 8 bit input dan 10 bit kunci). SDES didesain sebagai tes untuk mempelajari teknik kriptanalisis seperti *linear cryptanalysis, differential cryptanalysis,* dan *linear-differential cryptanalysis* pada *block chipher* (Vito,2002).

DES 8 putaran memiliki bagian-bagian yang sama seperti DES tetapi hanya dengan putaran yang lebih kecil yaitu 8 putaran.

Kriptanalis adalah setiap usaha untuk menemukan kunci atau menemukan *plaintext* dari *ciphertextnya*. *Differensial cryptanalisis* merupakan salah satu teknik kriptanalisis yang digunakan pada *block chiper*. *Differensial cryptanalisis* termasuk jenis *choses plaintext attack* yang berarti kriptanalis harus dapat mengenkripsi *plaintext* sesuai pilihannya.

Pada penelitihan skripsi ini digunakan metode *differensial cryptanalisis* untuk menemukan kunci serta lamanya waktu dalam melakukan *differensial cryptanalisis* pada DES.

1.2 Rumusan Masalah

Berdasarkan uraian latar belakang di atas, maka dapat dirumuskan beberapa permasalahan, antara lain:

1. Bagaimana mengimplementasikan *differensial cryptanalisis* pada SDES dan DES 8 putaran serta membuat aplikasi perangkat lunak yang mengimplementasikan *differensial cryptanalysis* .
2. Bagaimana mendapatkan *subkey* putaran terakhir pada SDES dan DES 8 putaran dengan menggunakan *differensial cryptanalysis.*

1.3 Batasan Masalah

Berdasarkan rumusan masalah yang telah disebutkan, maka pembahasan dibatasi pada :

1. Inputan file berupa file .txt dan .doc.
2. Inputan kunci berupa 1 karakter untuk SDES dan 8 karakter untuk DES 8 putaran.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah:

1. Mengimplemantasikan *differensial cryptanalisis* pada SDES dan DES 8 putaran dan membuat perangkat lunak yang dapat mengimplementasikan metode *differensial cryptanalisis*.
2. Memperoleh kunci putaran terakhir pada SDES dan DES 8 putaran dengan menggunakan *differensial cryptanalisis*.

1.5 Manfaat Penelitian

Manfaat yang dapat diambil dari pelaksanaan penelitian ini adalah :

1. Memberi gambaran mengenai metode *differensial cryptanalisis.*
2. Memberi gambaran untuk mendapatkan kunci putaran terakhir pada SDES dan DES 8 putaran dengan menggunakan *differensial cryptanalisis*.

1.6 Metodologi Penelitian

1. Studi Literatur
   Meliputi pemahaman tentang SDES dan DES 8 putaran dan *differensial cryptanalisis*.
2. Perancangan dan Pembuatan Program
   Membuat perancangan perangkat lunak dengan analisis terstruktur dan mengimplementasikan hasil rancangan tersebut dalam suatu program komputer.
3. Uji coba dan analisa hasil implementasi
   Menguji perangkat lunak, dan menganalisa hasil dari implementasi tersebut kemudian melakukan evaluasi.
4. Penyusunan Laporan
   Membuat laporan tertulis mengenai hasil tugas akhir ini.

1.7 Sistematika Penulisan

Tugas akhir ini disusun berdasarkan sistematika penulisan sebagai berikut:

1. **BAB I PENDAHULUAN**
   Berisi latar belakang masalah, perumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metodologi pemecahan masalah, dan sistematika penulisan.
2. **BAB II TINJAUAN PUSTAKA**
   Menguraikan teori-teori yang berhubungan dengan SDES, DES 8 putaran dan juga *differensial cryptanalisis*.
3. **BAB III METODOLOGI DAN PERANCANGAN SISTEM**
   Pada bab ini akan dijelaskan mengenai metode-metode yang digunakan dalam pembuatan perangkat lunak yang

mengimplementasikan differensial cryptanalisis pada SDES dan DES 8 putaran.

4. **BAB IV IMPLEMENTASI DAN UJI COBA SISTEM**
   Dalam bab ini dijelaskan mengenai implementasi program, uji coba dan analisanya
5. **BAB V KESIMPULAN DAN SARAN**
   Berisi kesimpulan dari seluruh rangkaian penelitian serta saran kemungkinan pengembangannya.

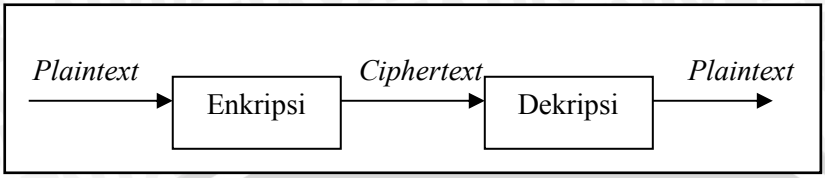
BAB II

TINJAUAN PUSTAKA

2.1 KRIPTOGRAFI

Kriptografi adalah ilmu yang mempelajari suatu cara agar dokumen atau pesan aman dan tidak bisa dibaca oleh pihak yang tidak berhak (Dony,2008). Ilmu ini sudah ada sejak zaman Mesir kuno. Sampai abad 20 kriptografi sangat penting dalam menentukan hasil perang. Pada tahun 1960 dengan adanya perkembangan teknologi komputer terdapat pemikiran serta permintaan untuk menerapkan kriptografi untuk melindungi data atau informasi dalam bentuk digital. Pada tahun 1977 diterapkannya DES (*Data Encryption Standard*) yang kemudian menjadi suatu standard keamanan data dibidang e-commerce. Pada tahun 1976 Diffie dan Helman mempublikasikan tulisan berjudul "*New direction in Cryptografi*" yang memperkenalkan revolusi pada kunci publik dan metode baru perubahan kunci kriptografi. Pada tahun 1978 Rivest, Shamir, dan Adleman menemukan enkripsi kunci publik dan skema tanda tangan yang disebut RSA. Tahun 1985 ditemukan skema kunci-publik oleh ElGamal yang didasarkan pada logaritma diskrit. Pada tahun 1991 standard internasional pertama untuk tandatangan digital telah disetujui (ISO/IEC 9796) yang didasarkan pada skema RSA kunci publik. Pada tahun 1994 pemerintah Amerika memakai standar tandatangan digital yang didasarkan pada skema kunci publik ElGamal (Budi,2004).

Enkripsi merupakan proses pengkodean sebuah pesan sehingga isi dari pesan tersebut tidak diketahui. Deskripsi adalah proses kebalikan dari enkripsi yaitu mentranformasi pesan yang dienkripsi kembali menjadi bentuk semula. Sebuah sistem enkripsi dan dekripsi disebut *cryptosystem.* Bentuk asli dari sebuah pesan disebut dengan *plaintext* dan bentuk asli yang dienkripsi disebut *ciphertext.* Hubungan tersebut digambarkan pada gambar 2.1.



Gambar 2.1 Enkripsi dan dekripsi pada *plaintext*
(Budi, 2004)

Plaintext dinotasikan dengan P dan *ciphertext* dinotasikan dengan C, secara matematika proses enkripsi ditulis dengan:

$$E(P) = C \quad (2.1)$$

dan untuk proses deskripsi dituliskan dengan rumus:

$$D(C) = P \quad (2.2)$$

(Budi, 2004).

Apabila fungsi enkripsi E ditambah sebuah kunci K maka notasi matematikanya menjadi

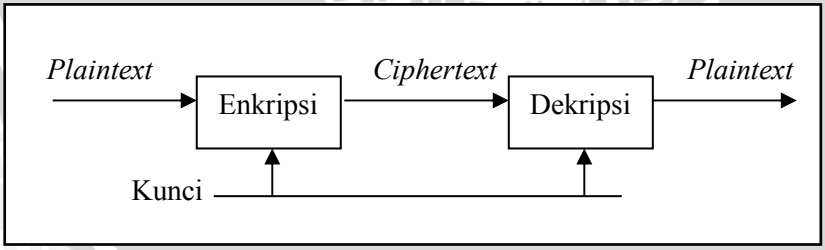
$$E_K(P) = C \quad (2.3)$$

dan untuk fungsi deskripsi D maka notasi matematikanya menjadi

$$D_K(C) = P \quad (2.4)$$

(Budi, 2004).

Untuk enkripsi dan deskripsi dengan sebuah kunci digambarkan pada gambar 2.2.



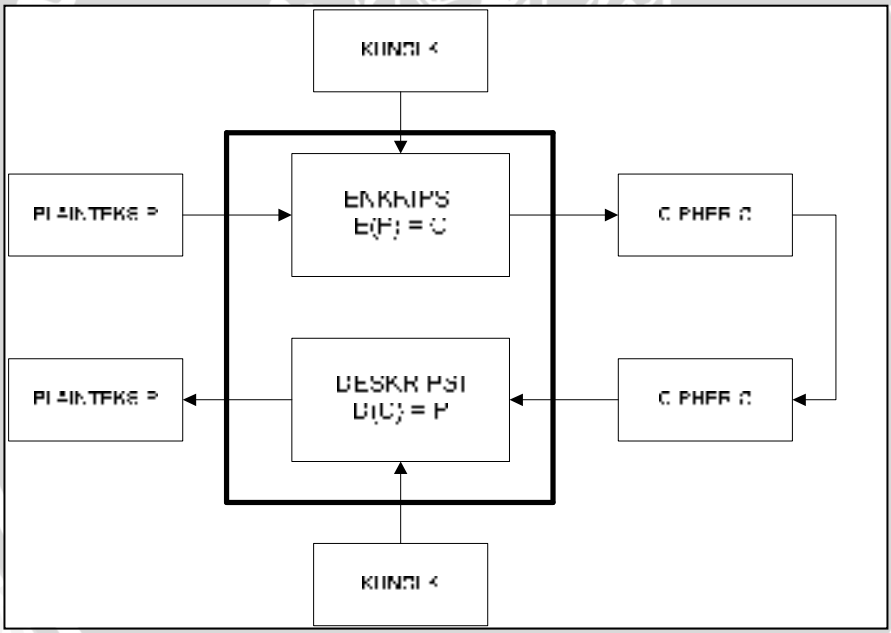
Gambar 2.2 Enkripsi dan deskripsi dengan sebuah kunci
(Budi, 2004)

2.2 ALGORITMA KRIPTOGRAFI

Algoritma kriptografi atau yang disebut *cipher* merupakan fungsi matematika yang digunakan untuk enkripsi dan deskripsi. Secara umum algoritma kriptografi dibedakan menjadi 2 berdasarkan kunci yang digunakan yaitu:

1. Algoritma kunci simetris

   Dalam sistem ini kunci yang digunakan untuk proses enkripsi dan dekripsi pada prinsipnya identik tetapi satu buah kunci dapat pula diturunkan dari kunci yang lainnya. Kunci-kunci pada algoritma ini harus benar-benar dirahasiakan oleh karena itu algoritma ini disebut algoritma kunci privat. Contoh algoritma ini adalah DES, Rijndael, Blowfish, IDEA, GHOST. Untuk hubungan enkripsi dan deskripsi pada algoritma kunci simetris digambarkan pada gambar 2.3.

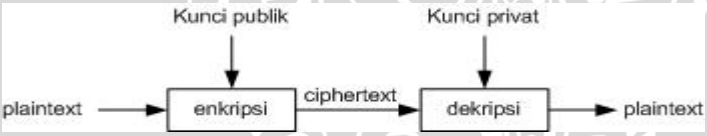


Gambar 2.3 Enkripsi dan deskripsi pada algoritma kunci simetris

1. Algoritma simetris dibagi menjadi dua yaitu *block cipher* dan *stream cipher*. *Block cipher* adalah metode enkripsi dimana *plaintext* dibagi-bagi dalam blok-blok string dengan panjang tertentu dan dienkripsi perblok. *Stream cipher* adalah metode enkripsi dimana blok string dibuat sama satu dengan yang lainnya dimana hal ini bermanfaat saat proses enkripsi dapat berubah pada tiap *plaintext* yang dienkripsi (www.smeapgri-tng.sch.id).

2. Algoritma kunci asymetris

Dalam algoritma ini digunakan 2 buah kunci. Satu kunci disebut kunci publik (*public key*) dimana kunci ini boleh dipublikasikan. Kunci yang satunya yaitu kunci private (*private key*) dimana kunci harus dirahasiakan. Contoh dari algoritma ini adalah RSA, PGP (www.smeapgri-tng.sch.id). Untuk hubungan enkripsi dan deskripsi pada algoritma kunci asimetris digambarkan pada gambar 2.4.



Gambar 2.4 Enkripsi dan deskripsi pada algoritma kunci asimetris

2.3 DES (*Data Encryption Standard*)

DES (*Data Encrytion Standard*) mengenkrip *plaintext* sebesar 64 bit (8 *byte*) dengan panjang kunci sekitar 56 bit (7 *byte*), sebanyak 16 putaran. Data 64 bit akan disubtitusi terlebih dahulu dengan permutasi IP (*initial permutation*). IP digunakan sebelum putaran pertama dari 16 putaran, dan mensubtitusi blok input dengan ketentuan seperti pada tabel 2.1.

Tabel 2.1 *Initial Permuted* (Budi, 2004)

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

Plaintext yang telah disubtitusi akan dipecah menjadi dua bagian sebesar 32 bit kiri **L** dan 32 bit kanan **R**. Pada setiap putarannya data kiri akan menjadi data kanan, sedangkan pada data kanan akan dilakukan operasi data kiri di-Xor-kan dengan fungsi f.

$$Li=Ri-1 \qquad (2.5)$$

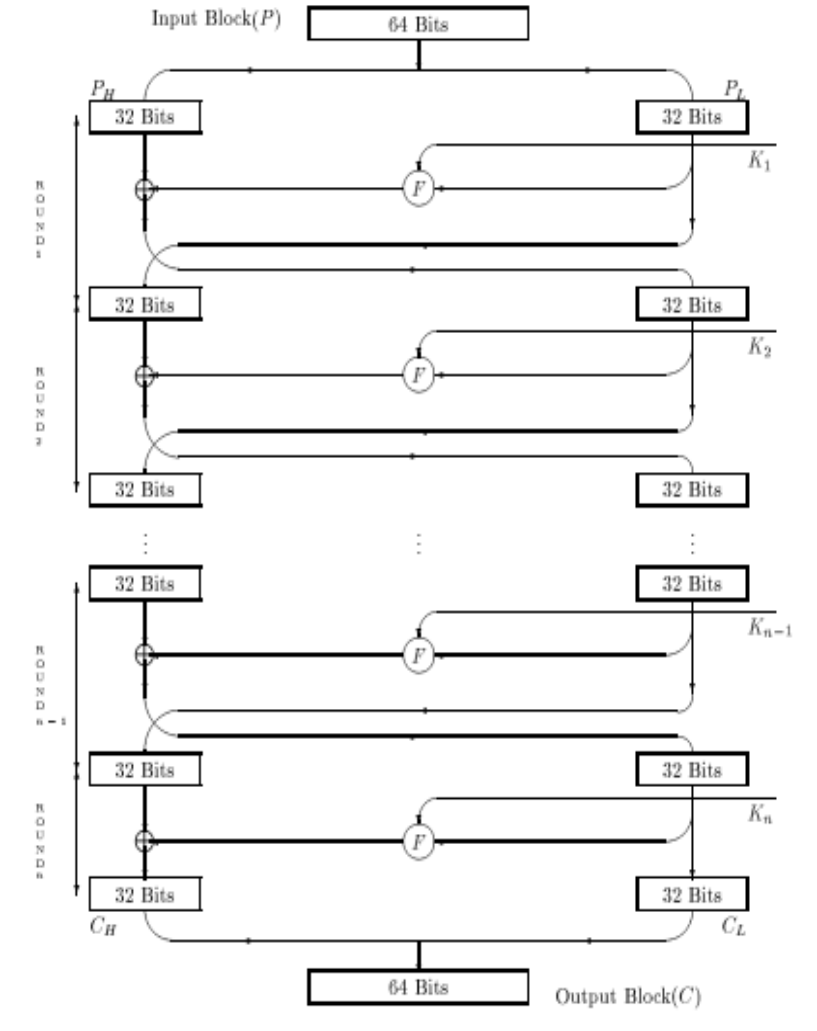
$$Ri=Li-1 \text{ Xor } f(Ri-1,Ki) \qquad (2.6)$$

Kemudian dilakukan subtitusi *final permutation atau* invers IP (IPinv). IPinv merupakan *invers* dari IP dan digambarkan dengan ketentuan seperti pada tabel 2.2.

Tabel 2.2 *Final Permutation* (Budi, 2004)

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

dalam putaran terakhirnya, blok $R_{16}L_{16}$ tidak terjadi pertukaran tetapi blok ini menjadi input untuk IPinv.



Gambar 2.5 Algoritma DES

Untuk fungsi f, data sebelah kanan sebesar 32 bit akan dipermutasi dengan *expansion permutation* (E) sehingga akan menghasilkan *ciphertext* sebesar 48 bit, kemudian dilakukan operasi Xor dengan blok kunci dan diinputkan ke dalam sbox. Sbox terdiri atas 8 buah. Hasilnya akan disubtitusi dengan P-Box *permutation* (P) sekaligus membentuk data menjadi 32 bit lagi. Isi dari E dan P dapat dilihat pada tabel 2.3 dan 2.4.

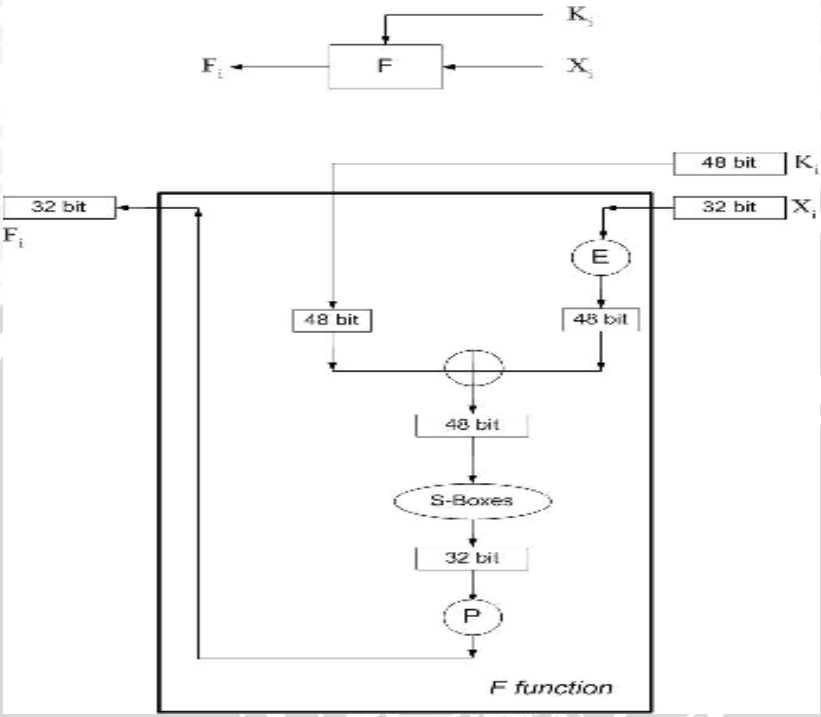
Tabel 2.3 *expansion permutation* (E) (Budi, 2004)

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

Tabel 2.4 P-Box *permutation* (P) (Budi, 2004)

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

Setiap putaran DES menggunakan fungsi F, dan pada kenyataannya keamanan DES tergantung pada fungsi ini. Fungsi f pada DES digambarkan pada gambar 2.6.



Gambar 2.6 Fungsi f pada DES

Untuk isi dari Sbox dapat dilihat pada tabel 2.5 sampai 2.12.

Tabel 2.5 S-Box 1 (Budi, 2004)

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

Tabel 2.6 S-Box 2 (Budi, 2004)

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

Tabel 2.7 S-Box 3 (Budi, 2004)

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

Tabel 2.8 S-Box 4 (Budi, 2004)

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

Tabel 2.9 S-Box 5 (Budi, 2004)

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

Tabel 2.10 S-Box 6 (Budi, 2004)

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

Tabel 2.11 S-Box 7 (Budi, 2004)

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

Tabel 2.12 S-Box 8 (Budi, 2004)

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

Blok kunci terdiri atas 16 blok, masing-masing berjumlah 56 bit. Inputan blok kunci (64 bit) akan didistribusi terlebih dahulu dengan *Permuted Choice 1* (Pc1) kemudian dipecah menjadi 2 bagian dengan setiap bagian sebesar 28 bit . Setiap blok ke 1, 2, 9, 16 akan dirotasi 1 bit ke kiri, sisanya 2 bit ke kiri kemudian digabungkan kembali menjadi 56 bit. Terakhir dilakukan permutasi dengan *Permuted Choice 2* (Pc2) sehingga menjadi 48 bit. Isi dari Pc1 dan Pc2 dapat dilihat pada tabel 2.13 dan 2.14.

Tabel 2.13 Pc1 (Budi, 2004)

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

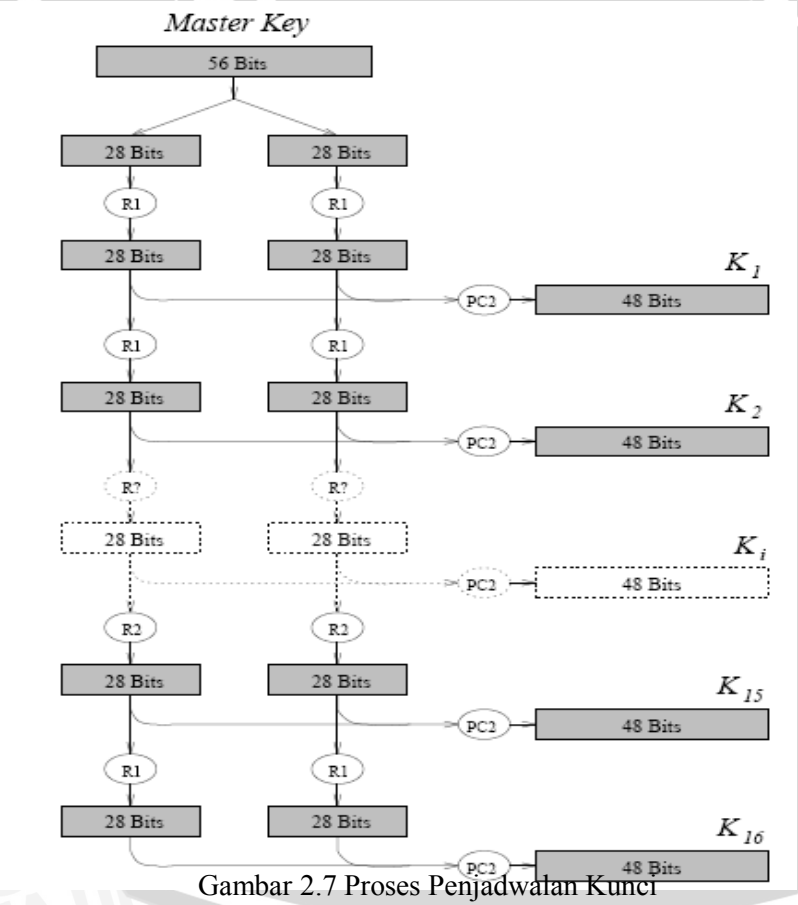
Tabel 2.14 Pc2 (Budi, 2004)

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

Proses penjadwalan kunci digambarkan pada gambar 2.7. Untuk iterasi pergeseran kekiri pada Cn dan Dn dituliskan pada tabel 2.15

Tabel 2.15 Tabel Iterasi

Putaran	Pergeseran	Putaran	Pergeseran
1	1	5	2
2	1	6	2
3	2	7	2
4	2	8	2



Gambar 2.7 Proses Penjadwalan Kunci

Untuk proses deskripsinya, operasinya akan dibalik dengan operasi enkripsi yaitu:

$$L_i = R_{i-1} \quad (2.7)$$

$$R_i = L_{i-1} \text{ Xor } f(R_{i-1}, K_i) \quad (2.8)$$

Dengan urutan kunci terbalik yaitu dimulai dari blok kunci yang terakhir.

2.4 SDES

SDES merupakan versi kecil dari algoritma DES. SDES hampir sama dengan DES tetapi dengan *block* dan ukuran kunci yang lebih kecil dari pada DES (dimana 8 bit *plaintext*/pesan dengan 10 bit kunci). SDES terdiri dari dua ronde (Vito,2002).

2.4.1 Kunci

10 bit kunci digunakan untuk menggenerate 2 *different block* dari 8 bit *subkey* dimana setiap *block* digunakan dalam iterasi. 10 bit kunci disebut *key*, 8 bit *subkey* disebut k1 dan k2. Untuk pembentukan *subkey*, *key* dimasukkan pada tabel permutasi yaitu tabel *permuted choise* 1 yang ditunjukkan pada tabel 2.16

Tabel 2.16 *Permuted Choise* 1 (Vito,2002)

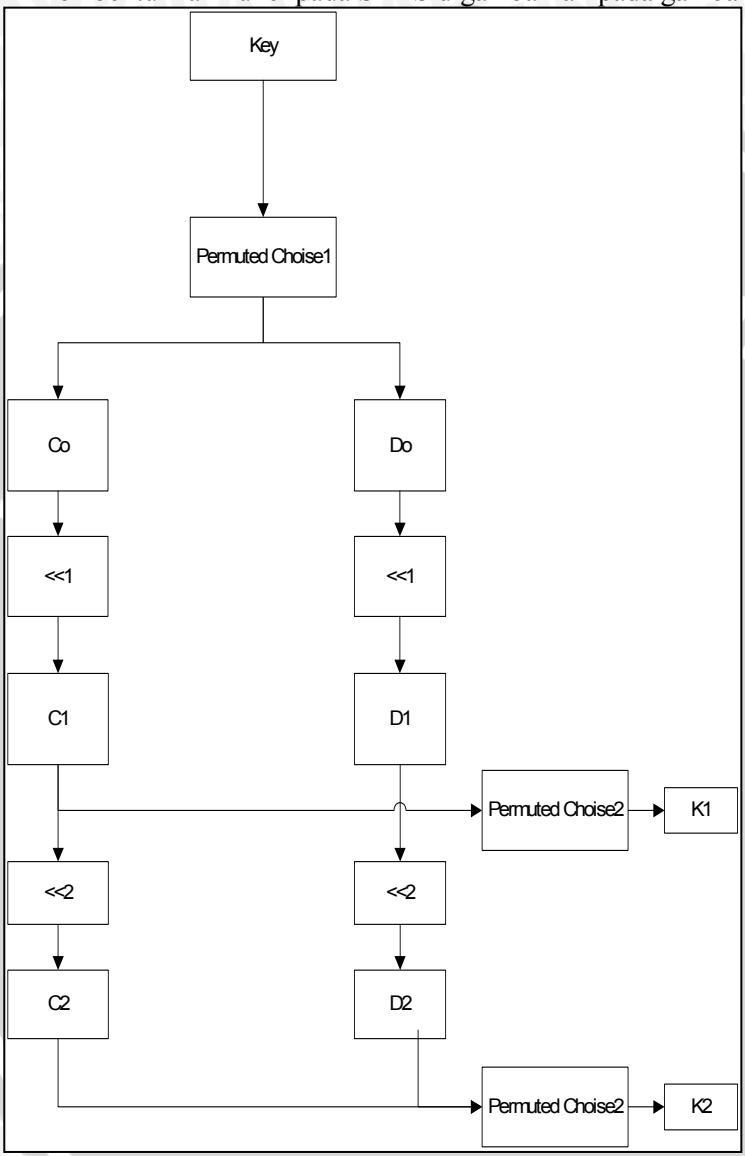
9	7	3	8	0
2	6	5	1	4

Tabel tersebut akan dibagi dua bagian. Bagian atas akan menjadi bit untuk Co dan bagian bawah akan menjadi bit untuk Do. Kemudian dilakukan pergeseran 1 bit kekiri pada Co dan Do yang kemudian hasilnya merupakan C1 dan D1. untuk menghasilkan K1, maka D1 digabungkan dengan C1 dan kemudian hasilnya merupakan masukan untuk tabel *permuted choise 2* dimana ditunjukkan pada tabel 2.17

Tabel 2.17 *Permuted Choise* 2 (Vito,2002)

3	1	7	5	0	6	4	2

Pembentukkan kunci pada SDES digambarkan pada gambar 2.8.



Gambar 2.8 Pembentukan kunci
(Vito,2002)

2.4.2 Enkripsi

Proses enkripsi pada SDES dapat ditulis dengan

$$C = E(P, K) = IP^{-1}(\rho 2(\rho 1(IP(P)))) \qquad (2.9)$$

8 bit *plainteks* menjadi masukan untuk *initial permutation*, IP1, yang ditunjukkan pada tabel 2.18

Tabel 2.18 IP 1 (Vito,2002)

7	6	4	0
2	5	1	3

Tabel tersebut akan dibagi dua bagian. Bagian atas akan menjadi bit untuk Lo dan bagian bawah akan menjadi bit untuk Ro. Setelah *initial permutation* Lo dan Ro akan memasuki putaran 1. Keluaran dari putaran 1 adalah L1 dan R1, yang dituliskan dengan

$$L1=Ro \qquad (2.10)$$

$$R1=Lo \text{ Xor } f(R,K1) \qquad (2.11)$$

L1 dan R1 akan memasuki putaran 2. Keluaran dari putaran 2 adalah L2 dan R2, yang dituliskan dengan

$$L2=R1 \qquad (2.12)$$

$$R2=L1 \text{ Xor } f(R,K2) \qquad (2.13)$$

L2 digabungkan dengan R2 dan kemudian hasilnya merupakan masukan untuk tabel *Inverse Initial Permutation* dimana ditunjukkan pada tabel 2.19

Tabel 2.19 IP^{-1} (Vito,2002)

3	6	4	7
2	5	1	0

Untuk fungsi f, data sebelah kanan sebesar 4 bit akan dipermutasi dengan *E bit selection* tabel (E) sehingga akan menghasilkan *ciphertext* sebesar 8 bit, kemudian dilakukan operasi Xor dengan blok kunci dimana K1 untuk putaran 1 dan K2 untuk putaran 2 dan diinputkan ke dalam sbox. Sbox terdiri atas 2 buah yaitu So dan S1. Hasil dari So dan S1 akan menjadi masukan untuk P. Untuk tabel E dapat dilihat pada tabel 2.20 dan untuk tabel So dan S1 dapat dilihat pada tabel 2.21 dan 2.22 dan Untuk tabel P dapat dilihat pada tabel 2.23

Tabel 2.20 E (Vito,2002)

3	0	1	2	1	2	3	0

Tabel 2.21 So (Vito,2002)

	So Colom			
Row	0	1	2	3
0	1	0	2	3
1	3	1	0	2
2	2	0	3	1
3	1	3	2	0

Tabel 2.22 S1 (Vito,2002)

	S1 Colom			
Row	0	1	2	3
0	0	3	1	2
1	3	2	0	1
2	1	0	3	2
3	2	1	3	0

Tabel 2.23 P (Vito,2002)

1	0	3	2

2.5 *CRYPTANALYSIS*

Cryptanalysis adalah ilmu untuk mendapatkan *plaintext* dari *ciphertext* tanpa memiliki kunci untuk membuka *ciphertext* tersebut. Orang yang melakukan *cryptanalysis* disebut kriptoanalis (*cryptanalysts*). Sedangkan usaha untuk melakukan *cryptanalysis* disebut serangan (*attack*) (Schneier, 1996). Berdasarkan ketersediaan data, serangan dapat dikelompokkan menjadi:

1. *Ciphertext-only attack*
   Kriptanalis memiliki beberapa *ciphertext* dari beberapa pesan, semuanya dienkripsi dengan algoritma yang sama.
2. *Known-plaintext attack*
   Beberapa pesan yang formatnya terstruktur membuka peluang kepada kriptanalis untuk menerka *plaintext* dari *ciphertext* yang bersesuaian.
3. *Chosen-plaintext attack*
   Serangan jenis ini lebih hebat daripada *known plaintext attack*, karena kriptanalis dapat memilih *plaintext* tertentu untuk dienkripsikan, yaitu *plaintext-plaintext* yang lebih mengarahkan penemuan kunci.
4. *Adaptive-chosen-plaintext attack*
   Kasus khusus dari jenis serangan nomor 3 di atas. Misalnya kriptanalis memilih *blok plaintext* yang besar, lalu dienkripsi, kemudian memilih blok lainnya yang lebih kecil berdasarkan hasil serangan sebelumnya, begitu seterusnya.
5. *Chosen-ciphertext attack*
   Kriptanalisis memiliki akses terhadap *ciphertext* yang dideskripsi.
6. *Chosen-text attack*
   Gabungan *chosen-plaintext attack* dan *chosen-ciphertext attack.*

2.6 *DIFFERENSIAL CRYPTANALYSIS*

Differensial cryptanalysis merupakan salah satu teknik kriptanalisis yang digunakan pada *block cipher*. *Differensial*

cryptanalysis termasuk jenis *chosen plaintext attack*, yang berarti bahwa kriptanalis harus dapat mengenkripsi *plaintext* sesuai dengan pilihannya. Cara kerja dari metode ini adalah dengan menganalisa perkembangan dari perbedaan (*difference*) hasil enkripsi pasangan *plaintext* dengan menggunakan kunci yang sama.

Differensial cryptanalysis didasarkan pada observasi dalam jumlah besar dari perbedaan *plaintext* dan perbedaan sampai putaran terakhir dari *cipher*. *Difference* dari input dapat dituliskan dengan

$$\Delta X = X' \oplus X'' \qquad (2.11)$$

dimana $\oplus$ merupakan komponen dari wise XOR

$$\Delta X = [\, \Delta X_1 \, \Delta X_2 \ldots \Delta X_n \,] \qquad (2.12)$$
$$\Delta X_i = X_i' \oplus X_i'' \qquad (2.13)$$

Dimana X_i' dan X_i'' merepresentasikan bit ke-i dari X_i' dan X_i''. Untuk *difference* dari output dapat ditulis

$$\Delta Y = Y' \oplus Y'' \qquad (2.14)$$
$$\Delta Y = [\, \Delta Y_1 \, \Delta Y_2 \ldots \Delta Y_n \,] \qquad (2.15)$$
$$\Delta Y_i = Y_i' \oplus Y_i'' \qquad (2.16)$$

Keuntungan dari kriptanalisis deferensial kemudahan untuk memprediksi *output difference* dari operasi linear yang diberikan *input difference:*

1. *Operasi Unary* (E,P,IP)

$$(P(X))` = P(X) + P(X^*) = P(X`) \qquad (2.12)$$

2. *Operasi Binary*

$$(X + Y)` = (X + Y) + (X^* + Y^*) = X`+Y` \qquad (2.13)$$

3. *Mixing the key*

$$(X + K)` = (X + K) + (X^* + K) = X` \qquad (2.14)$$

BAB III
METODOLOGI DAN PERANCANGAN

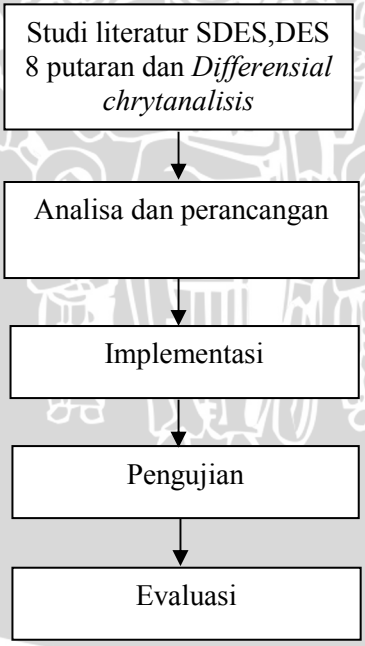
Dalam bab ini dijelaskan tentang metode dari perancangan perangkat lunak untuk mengimplementasikan differensial chriptanalisis yang dilakukan pada algoritma kriptografi DES.

3.1 Identifikasi Perangkat Lunak

Perangkat lunak yang akan dibangun merupakan implementasi dari *differensial cryptanalisis* yang dilakukan pada DES. Secara umum, langkah-langkah dalam pembuatannya adalah

1. Studi literatur tentang SDES, DES 8 putaran dan *differensial chryptanalisis* pada SDES dan DES 8 putaran
2. Analisa dan perancangan perangkat lunak
3. Implementasi perangkat lunak
4. Pengujian
5. Evaluasi

Langkah-langkah pembuatan ditunjukkan dalam Gambar 3.1



Gambar 3.1 Langkah-langkah pembuatan perangkat lunak

3.1.1 Deskripsi Umum Perangkat Lunak

Perangkat lunak yang dibuat merupakan implementasi dari teori *differensial cryptanalisis* dimana perangkat lunak ini digunakan untuk menemukan bit-bit kunci yang digunakan dalam proses enkripsi *plaintext* menjadi *chipertext* pada DES. Untuk itu dilakukan analisis terhadap *difference* dari *plaintext* dan *chipertext* yang dihasilkan dari enkripsi yang dilakukan DES untuk menemukan bit-bit kunci yang digunakan pada proses enkripsi tersebut. Perangkat lunak yang dibuat dapat melakukan enkripsi pada *plaintext* yang ditentukan dengan menggunakan algoritma DES serta melakukan *differensial cryptanalisis* pada DES dengan melakukan analisa *difference* dari *plaintext* dan *chipertext* yang dihasilkan dari proses enkripsi.

Input dari perangkat lunak ini adalah pilihan algoritma yang digunakan yaitu SDES dan DES 8 putaran, kunci dimana besarnya kunci ditentukan dari algoritma DES yang dipilih, untuk SDES kunci yang diinputkan adalah sejumlah 10 bit dan untuk DES 8 putaran sebesar 56 bit, jumlah *plainteks* yang diinginkan. Kunci digunakan untuk mengenkripsi *plainteks.*

Proses *differensial cryptanalysis* itu sendiri dilakukan terhadap pasangan *plaintext* dan *ciphertext*-nya, yang akan menghasilkan sub*key* dari kunci aslinya.

Ketika *user* memasuki perangkat lunak, proses yang terjadi adalah:

1. Perangkat lunak meminta inputan berupa pilihan algoritma SDES atau DES 8 putaran, kunci sesuai algoritma yang dipilih 10 bit untuk SDES dan 64 bit untuk DES 8 putaran, file yang merupakan *plainteks* yang akan dienkripsi.
2. *Plaintext* dienkripsi menggunakan algoritma kriptografi DES yang dipilih dengan parameter berupa kunci yang telah diinputkan sebelumnya, untuk memperoleh pasangan *ciphertext*-nya. *Plaintext* dan *ciphertext* tersebut digunakan dalam proses *differensial cryptanalyisis*.
3. Perangkat lunak kemudian melakukan *differensial cryptanalysis* terhadap *plaintext* dan *ciphertext* tadi.
4. Proses *differensial cryptanalysis* menghasilkan *subkey.*
5. *Subkey* yang ditemukan akan diuji untuk mendapatkan kunci putaran terakhir kemudian dicocokan dengan *subkey* yang asli.

Kunci pada putaran terakhir dari SDES atau DES 8 putaran yang akan menjadi output dari perangkat lunak ini.

3.1.2 Batasan Perangkat Lunak

Batasan dari perangkat lunak yang akan dibuat adalah:

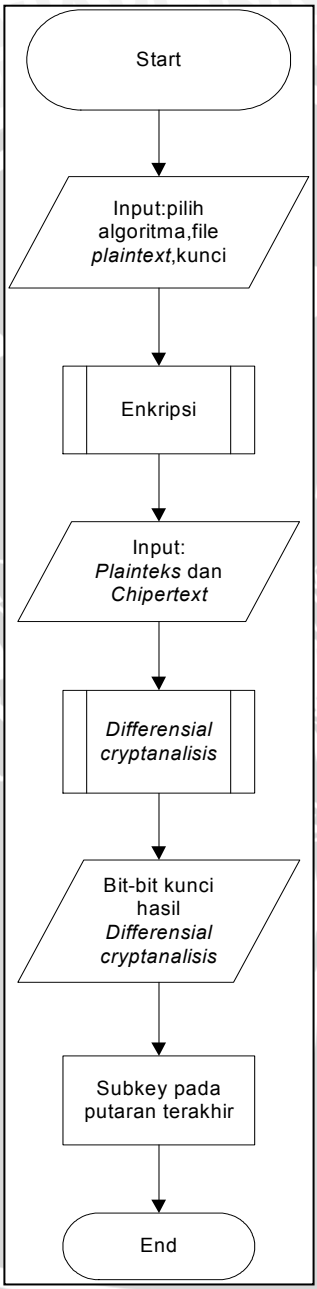
1. Perangkat lunak hanya bisa mengenkripsi tidak dibuat proses untuk deskripsi.
2. Inputan berupa file yang digunakan sebagai *plainteks*.

3.2 Perancangan Perangkat Lunak

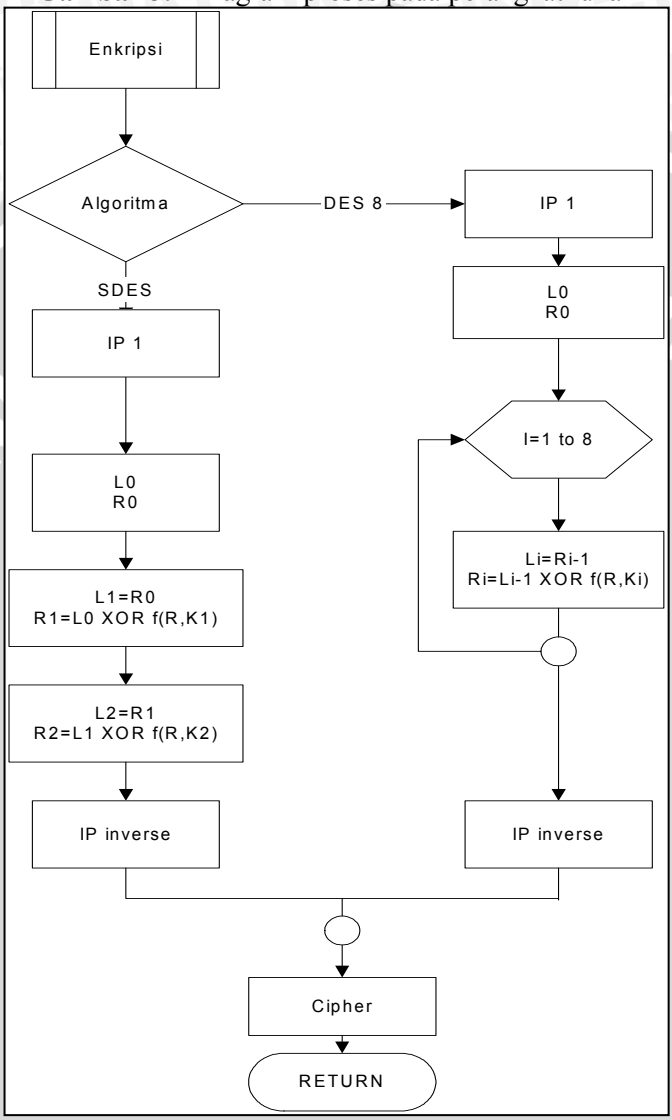
Secara keseluruhan, langkah-langkah pembuatan perangkat lunak yang mengimplementasikan *differensial cryptanalisis* pada DES dapat dilihat pada Gambar 3.2 dan arsitektur serta proses yang terjadi pada perangkat lunak yang akan dibangun akan dibahas berikut ini.

3.2.1 Proses Input

Inputan dari proses ini terdiri dari tiga yaitu inputan pilihan algoritma yang digunakan (SDES dan DES 8 putaran), file *plaintext* yang diinginkan (1 *plaintext* = 16 bit untuk DES 8 putaran dan 4 bit enkripsi untuk SDES), kunci yang nanti akan digunakan dalam proses dimana jumlah kunci yang dimasukkan sesuai dengan algoritma yang dipilih (10 bit untuk SDES dan 64 bit untuk DES 8 putaran). Gambar 3.2 merupakan proses yang terjadi pada sistem secara umum dimana dimasukkan inputan file, algoritma serta kunci kemudian dilakukan proses enkripsi dihasilkan cipherteks yang kemudian plainteks dan cipherteks ini menjadi inputan untuk *differensial cryptanalysis* dimana akan dihasilkan *subkey* pada putaran terakhir.


Start
Input:pilih algoritma,file *plaintext*,kunci
Enkripsi
Input: *Plainteks* dan *Chipertext*
Differensial cryptanalisis
Bit-bit kunci hasil *Differensial cryptanalisis*
Subkey pada putaran terakhir
End

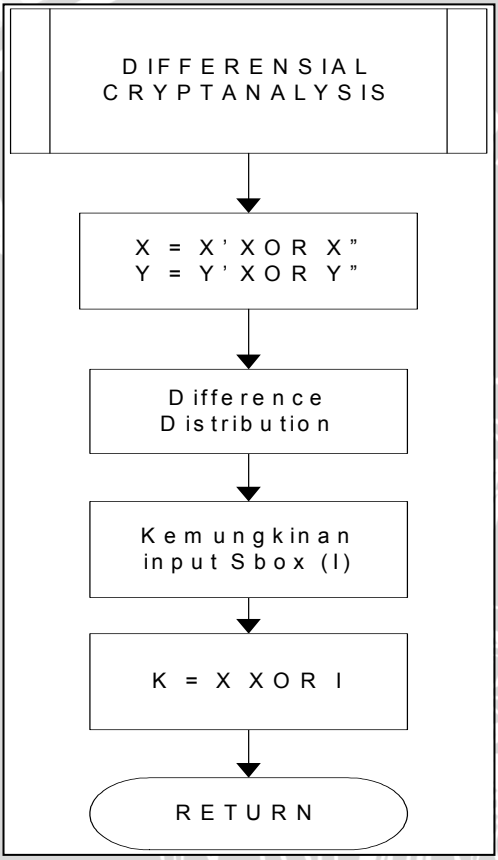
Gambar 3.2 Diagram proses pada perangkat lunak



Gambar 3.3 Diagram proses Enkripsi

Gambar 3.3 menunjukkan proses enkripsi dimana plainteks yang telah diubah dalam biner dimasukkan kedalam tabel IP1 kemudian

dipecah menjadi L0 dan R0 yang akan dilakukan proses sebanyak putaran hasilnya dimasukkan dalam IP inverse yang menghasilkan cipherteks.



Gambar 3.4 Diagram proses Differensial Cryptanalysis

Gambar 3.4 merupakan proses *differensial cryptanalysis* dimana dicari ΔX dari *plainteks* dan ΔY dari *cipherteks* kemudian dicari kemungkinan inputan untuk Sbox dari *difference distribution* tabel. Dilakukan XOR kemungkinan

inputan Sbox dengan plainteks yang menghasilkan kemungkinan kunci.

3.2.2 Enkripsi

Proses enkripsi ini menggunakan algoritma yang telah dipilih serta kunci yang dimasukkan untuk mengenkripsi *plainteks* menjadi *chiperteks*. Proses enkripsi secara teori telah dijelaskan pada sub bab 2.3 dimana hasil dari *plainteks* dan *chiperteks* tersebut akan digunakan untuk proses *differensial cryptanalisis* untuk menemukan bit-bit kunci.

3.2.3 Differensial Cryptanalisis

Dengan menggunakan *differensial distribution table* pada S-box dimana baris untuk ΔX dan kolom untuk ΔY, didapat dua kemungkinan yaitu:

1. Dapat diperoleh input dan output yang mungkin dengan diketahui *difference*nya. Hal ini dilakukan dengan mengecek *difference distribution table.*
2. Dapat diketahui bit kunci melalui S-box dengan menggunakan pasangan input/*plainteks* yang telah diketahui dan *difference* output pada S-box.

   Contoh: X'= 2, X"=8 dan S-box merupakan So. Kemudian untuk Y'= 0 dan Y"= 2 dan $\Delta Y=2$ dimana $\Delta X=\Delta I=2\oplus 8=10$, dimana pasangan yang cocok untuk *difference* tersebut adalah 7 dan 13. Untuk $K=I\oplus X$, didapat kemungkinan 1 kunci adalah 5 dihitung dari:

$$K = I' \oplus X' = 7 \oplus 2 = 5 \text{ dan } K = I' \oplus X'' = 7 \oplus 8 = 5$$

kemungkinan 2 kunci adalah 15 dihitung dari:

$$K = I' \oplus X' = 13 \oplus 2 = 15 \text{ dan } K = I' \oplus X'' = 13 \oplus 8 = 15$$

Tabel 3.1 contoh kemungkinan kunci

S-box Input	Possible Key
7 13	5 15

Dari perhitungan diatas dapat digunakan untuk menganalisa untuk membuat *difference characteristic* dimana dengan menggunakan *difference characteristic* dapat didapatkan *subkey* pada putaran terakhir.

Proses ekstrak *subkey* dideskripsikan dengan algoritma sebagai berikut:

1. Masukan *plainteks*.
2. Untuk P' $\oplus$ Δ X = P"
    a. Enkripsi P' dan P" dengan kedua kunci K1 dan K2 untuk menghasilkan C' dan C", juga hasilkan C1' dan C1" dengan mengenkripsi *plainteks* setelah putaran 1 dimana enkripsi hanya dilakukan dengan K1.
    b. Jika C1' $\oplus$ C1" = ΔY, lalu
        i. Untuk semua *subkey* yang mungkin enkripsi C1'$\oplus$C1" dengan hanya putaran kedua
        ii. Jika hasil enkripsi dari 2.b.i sama dengan C' dan C" maka naikkan nilai dari *subkey* yang digunakan.
3. Ulangi langkah 1 dan 2 untuk *plainteks* yang lainnya. *Subkey* yang memiliki nilai lebih dari *subkey* yang lainnya merupakan *subkey* yang benar.

3.3 Perancangan Uji Coba

Setelah proses perancangan perangkat lunak selesai dilakukan, maka dilakukan proses uji coba dari perangkat lunak tersebut dengan langkah-langkah sebagai berikut :

1. Mulai
2. Masukan file inputan.
3. Enkripsi.
4. *Differensial Cryptanalisis*.
5. Pengujian *subkey*.
6. Pencocokan kunci putaran terakhir yang ditemukan dengan kunci putaran terakhir asli.
7. Selesai.

3.3.1 Pencocokan Kunci yang ditemukan dengan kunci asli

Untuk pencocokan *subkey* yang didapat dengan yang asli serta lama waktu dibuat pada tabel 3.2 dan 3.3

Tabel 3.2 Hasil uji coba *differensial cryptanalisis* SDES

Kunci	Subkey	Subkey'	count

Keterangan:

- Kunci merupakan kunci asli yang dimasukkan untuk enkripsi
- Subkey merupakan Subkey pada putaran terakhir
- Subkey' merupakan subkey yang dihasilkan dari *differensial cryptanalisis*.
- count merupakan banyak iterasi pada percobaan.

Tabel 3.3 Hasil uji coba *differensial cryptanalisis* DES 8 putaran

Kunci	Subkey	Subkey'	count

Keterangan:

- Kunci merupakan kunci asli yang dimasukkan untuk enkripsi
- Subkey merupakan Subkey pada putaran terakhir
- Subkey' merupakan subkey yang dihasilkan dari *differensial cryptanalisis*.
- count merupakan banyak iterasi pada percobaan.

Untuk mendapatkan kunci dari beberapa kemungkinan kunci dibuat tabel 3.4

Tabel 3.4 tabel kemungkinan kunci putaran terakhir

File	Kunci	Count

Keterangan:

- File merupakan nama file yang diambil untuk *enkripsi* dan uji.
- Kunci merupakan kemungkinan kunci pada putaran terakhir
- Count merupakan banyak iterasi pada percobaan..

3.4 Perancangan antar muka

Form awal (Gambar 3.3) terdiri dari beberapa bagian antara lain:

1. Masukan kunci sesuai algoritma yang diinginkan.
2. Masukan file *plainteks* dan tombol browse.
3. Pilihan algoritma yang digunakan untuk enkripsi
4. Tombol untuk proses, hasil.

Form hasil (Gambar 3.4) terdiri dari tiga bagian yaitu:

1. Kunci asli.
2. Kunci putaran terakhir asli.
3. Kemungkinan kunci yang ditemukan.

3.5 Contoh perhitungan

Pada subbab ini akan ditunjukkan contoh perhitungan sederhana untuk proses enkripsi pada SDES dan DES 8 putaran serta proses *differensial cryptanalisis*.

1. Pembentukan kunci pada SDES

Kunci yang digunakan pada SDES adalah 10 bit yang dimasukkan ke dalam PC1 yang terdapat pada tabel 2.16.

Contoh 10 bit kunci:

Kunci = a

K =0001100001

Kemudian didapat 10 bit kunci hasil permutasi.

K+=1010000001

Kemudian kunci hasil permutasi tadi dipecah menjadi dua bagian

Co = 10100

Do = 00001

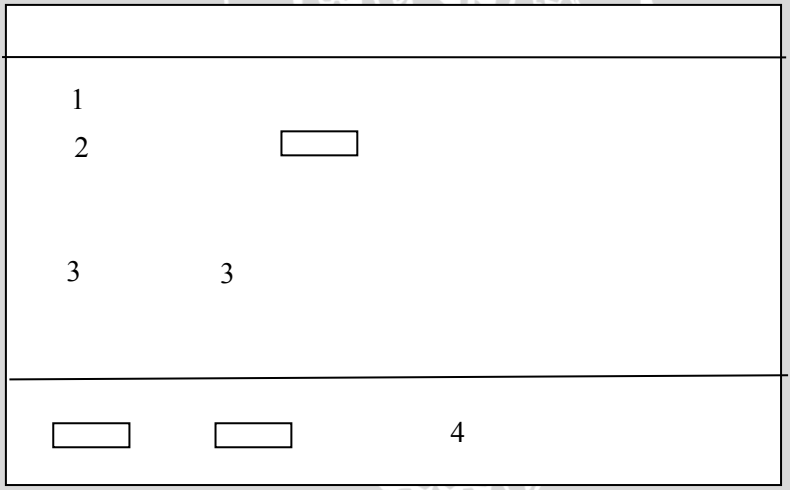
dengan Co dan Do akan dibentuk C1 dan D1 untuk menghasilkan K1 serta C2 dan D2 untuk menghasilkan K2. untuk menghasilkan C1 dan D1dilakukan pergeseran 1 bit kekiri pada Co dan Do dan untuk menghasilkan C2 dan D2 dilakukan pergeseran 2 bit kekiri pada C1 dan D1. Contoh:

Co = 10100
Do = 00001

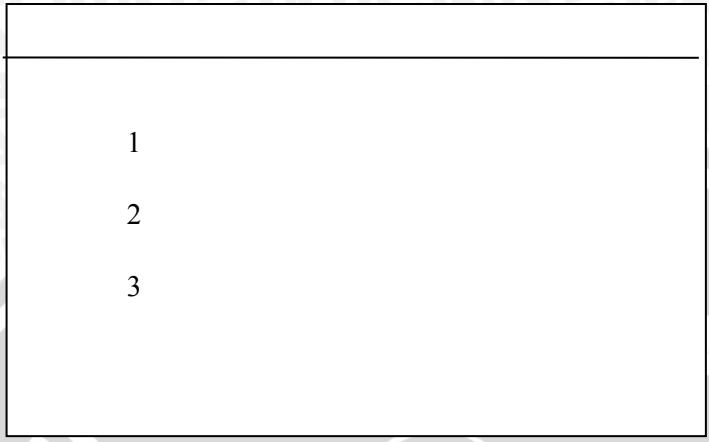
C1 = 01001
D1 = 00010

C2 = 00101
D2 = 01000

Untuk kunci 1 didapatkan dari C1D1 = 0100100010



Gambar 3.5 *Form* awal



Gambar 3.6 *Form* Hasil

Untuk kunci 2 didapatkan dari C2D2 = 0010101000

Kemudian CnDn masukan untuk PC2 pada tabel 2.17 untuk mendapatkan K1dan K2 sebesar 8 bit.

K1= 01000010
K2= 00000111

2. Pembentukan kunci pada DES 8 putaran

Kunci yang digunakan pada DES 8 putaran adalah 64 bit dimana bit ke 8 merupakan parity bit sehingga kunci yang digunakan menjadi 56 bit kunci. Kunci tersebut menjadi masukkan ke dalam PC1 yang terdapat pada tabel 2.13.
Contoh:

Kunci : annoradc

K = 01100001 01101110 01101110 01101111 01110010 01100001 01100100 01100011

K+ = 0000000 0111111 1111111 1110001 1001111 0010011 1000001 1100000

Kemudian kunci hasil permutasi tadi dipecah menjadi dua bagian

C_0 = 0000000 0111111 1111111 1110001
D_0 = 1001111 0010011 1000001 1100000

Dengan C_0 dan D_0 maka dibentuk 8 blok dengan menggeser kekiri dimana pergeseran pada tiap putarannya terdapat pada tabel 2.15

C_0 = 00000000111111111111111110001
D_0 = 10011110010011100000111000000

C_1 = 00000001111111111111111100010
D_1 = 00111100100111000001110000001

C_2 = 00000011111111111111111000100
D_2 = 01111001001110000011100000010

C_3 = 00001111111111111111100010000
D_3 = 11100100111000001110000001001

C_4 = 00111111111111111110001000000
D_4 = 10010011100000111000000100111

C_5 = 11111111111111111000100000000
D_5 = 01001110000011100000010011110

C_6 = 11111111111111100010000000011
D_6 = 00111000000111000001001111001

C_7 = 11111111111110001000000001111
D_7 = 11100000111000001001111001000

C_8 = 11111111110001000000000111111
D_8 = 10000011100000100111100100111

Kemudian CnDn masukan untuk PC2 pada tabel 2.14 untuk mendapatkan Kn.

K_1= 111000 001011 111001 101110 101100 010011 010110 011001
K_2= 111000 001011 011011 110110 101011 110011 011000 000101
K_3= 111101 001101 011001 110110 001110 100100 001111 100010
K_4= 111001 101101 001101 110010 000101 001100 100100 000111
K_5= 101011 101101 001101 110111 110001 100010 010011 010000
K_6= 101011 110101 001101 011011 111010 011010 001101 001001
K_7= 001011 110101 001111 111001 001100 101101 011000 001010
K_8= 100111 110101 100111 011001 010111 000001 010100 100110

3. Enkripsi pada SDES

Enkripsi pada SDES dimana terdapat sebuah pesan M sebesar 8 bit merupakan inputan untuk IP1 yang terdapat pada tabel 2.18.

Contoh : *Plainteks* = halo
M = 01101000
IP1= 00101010

Dari IP1 didapat Lo dan Ro yaitu
Lo = 0010
Ro = 1010

Dengan menggunakan fungsi f dan kunci yang didapat Ln dan Rn.

Contoh : n = 1
K1 = 01000010
L1 = Ro = 1010
R1 = Lo Xor f(R,K1)

Disini fungsi f bekerja. Untuk menghitung fungsi f maka pertama Rn-1 akan diexpand dengan menggunakan tabel 2.20 dimana akan membuat 4 bit R menjadi 8 bit.

Ro = 1010
E(Ro) = 01010101

Kemudian menghitung fungsi f dengan mengXor-kan E(Ro) dengan K1.

Contoh:
K1 = 01000010
E(Ro) = 01010101
K1+E(Ro) = 00010111

Kemudian hasilnya dimasukan kedalam S-box pada tabel 2.21 dan 2.21 kemudian hasil keluaran S-box dilakukan permutasi menggunakan tabel P yang terdapat pada tabel 2.23.

f = 1110

Kemudian pada hasil f dilakukan Xor dengan Lo untuk menghasilkan R1.

R1 = 1100

Untuk L2 dan R2 dilakukan hal yang sama yaitu

K2 = 00000111
L2 = R1 = 1100
R2 = L1 Xor f(R,K2) = 0111

Setelah putaran terakhir ini maka dilakukan IP invers yang terdapat pada tabel 2.19 untuk menghasilkan *chipertext*

L2R2 = 11000111
C = 01010111

Hasil *chipertxt*nya adalah C = 01010111

4. Enkripsi pada DES 8 putaran

Enkripsi pada DES 8 putaran dimana terdapat sebuah pesan M sebesar 64 bit merupakan inputan untuk IP yang terdapat pada tabel 2.1. Contoh :

Plainteks = halo apa kabar?
M = 0110 1000 0110 0001 0110 1100 0110 1111 0010 0000 0110 0001 0111 0000 0110 0001

IP = 1110 1111 0100 0000 0000 1100 1010 1010 0000 0000 1111 1111 0000 1101 0000 1000

Didapat Lo dan Ro yaitu

L_0 = 1110 1111 0100 0000 0000 1100 1010 1010
R_0 = 0000 0000 1111 1111 0000 1101 0000 1000

Dengan menggunakan fungsi f dan kunci yang didapat Ln dan Rn.
Contoh :

K_1= 111000 001011 111001 101110 101100 010011 010110 011001
$L_1 = R_0$ = 0000 0000 1111 1111 0000 1101 0000 1000
$R_1 = L_0 + f(R_0,K_1)$

Disini fungsi f bekerja. Untuk menghitung fungsi f maka pertama Rn-1 akan diexpand dengan menggunakan tabel 2.3

R_0 = 0000 0000 1111 1111 0000 1101 0000 1000
$E(R_0)$ = 000000 000001 011111 111110 100001 011010 100001 010000

Kemudian menghitung fungsi f dengan mengXor-kan E(Ro) dengan K1.
Contoh:

K_1 = 111000 001011 111001 101110 101100 010011 010110 011001
$E(R_0)$ = 000000 000001 011111 111110 100001 011010 100001 010000
$K_1+E(R_0)$ = 111000 001010 100110 010000 001101 001001 110111 001001.

Kemudian hasilnya dimasukan kedalam S-box pada tabel 2.5 sampai 2.12 kemudian hasil keluaran S-box dilakukan permutasi menggunakan tabel P yang terdapat pada tabel 2.4.

f = 1110 1111 0011 1110 0110 0111 0000 1011
R_1 = 0000 0000 0111 1110 0110 1011 1010 0001

Selanjutnya dilakukan sampai didapat L_8 dan R_8.
L_8 R_8= 01001110 10101011 10000011 01011010 11000001 01011011 11100111 00000001

Kemudian hasil L_8 R_8 dimasukkan ke dalam IP invers pada tabel 2.2 untuk menghasilkan *chipertext*nya.

C = 10111110 01111101 01001000 01110001 00100001 00011000 11101001 10011100

5. *Differensial cryptanalisis* pada SDES

Differensial cryptanalisis ini dilakukan untuk memperoleh *subkey* K2 pada SDES sehingga diperoleh 8 bit kunci dari 10 bit kunci aslinya. Mula-mula dari 2 *plaintext* dicari ΔXo, ΔYo yaitu *difference* dari X' dan X" dari So dan juga ΔX1, ΔY1 yaitu *difference* dari X' dan X" dari S1 pada putaran 2. Contoh:

Mula-mula dari 2 *plaintext* dicari ΔXo, ΔYo, ΔX1, ΔY1 pada putaran kedua.

S0:

X' = 0110	X" = 0011
Y' = 0011	Y" = 0001

S0 : ΔXo = 0101
ΔYo = 0010

S1:

X' = 1001	X" = 1100
Y' = 0010	Y" = 0001

S1 : ΔX1 = 0101
ΔY1 = 0011

Untuk mencari subkey K2 maka dalam perhitungan pencarian kunci yang mungkin Subkey K2 dibagi menjadi 2 bagian kiri dan kanan dimana setelah diXORkan dengan E(Ro) hasilnya merupakan masukan untuk Sbox, dimana bagian kiri merupakan masukan untik So dan bagian kanan merupakan masukan untuk S1 dan hal ini dilakukan hanya pada putaran kedua.

Pada putaran kedua dicari kemungkinan masukan yang menghasilkan keluaran pada S0 dan S1 maka didapat angka-angka yaitu

S0	S1
0110	0110
0001	0011
1100	1110
1011	1001

Kemudian dipilih masukan yang benar dengan cara membandingkan hasil XOR kemungkinan masukan dengan ΔXo untuk S0 dan ΔX1 untuk S1 dimana didapat kemungkinan masukan yang sesuai adalah

S0	S1
0110	1110
0011	1011
1100	
1001	
1011	
1110	

Untuk mendapatkan kemungkinan kunci maka hasil kemungkinan masukan untuk S0 dan S1 diXOR dengan masukan sebelum XOR dengan kunci didapat kemungkinan kunci yaitu:

Perhitungan kemungkinan kunci pada So:

0000
0101
1010
1111
1101
1000

Sehingga dihasilkan kemungkinan kunci untuk K2 bagian kiri yaitu 0000, 0101, 1010, 1111, 1101, 1000.

Untuk mendapatkan kemungkinan kunci pada bagian kanan dilakukan hal yang sama.

Perhitungan kemungkinan kunci pada S1:

0111
0010

Sehingga dihasilkan kemungkinan kunci untuk K2 bagian kanan yaitu 0111, 0010.

Sehingga didapat kemungkinan K2 yaitu

00000111	00000010
01010111	01010010
10100111	10100010
11110111	11110010
11010111	11010010
10000111	10000010

Untuk *differensial* pada 8 putaran juga dilakukan hal yang sama pada putaran terakhirnya untuk mendapatkan kemungkinan kunci K8.

6. Pengujian kemungkinan kunci pada SDES

Kemungkinan kunci yang telah didapat tadi diuji untuk mendapatkan kunci yang benar pada putaran terakhir. Pada putaran 2 semua kemungkinan kunci dicoba untuk mendapatkan kunci yang cocok. Pada contoh perhitungan ini hanya dicobakan pada empat kemungkinan kunci saja yaitu 00000111, 01010111, 10100111, 11110111.

K: 00000111	K: 01010111
E(R1) = 0110 1001 K2 = 00000111 01101110	E(R1) = 0110 1001 K2 = 01010111 00111110

Sbox = 1110 f = 1101 L1 = 1010 R2 = 0111 L2R2 = 1100 0111 *C' = 01010111 C' = 01010111	Sbox = 0110 f = 1001 L1 = 1010 R2 = 0011 L2R2 = 1100 0011 *C' = 01010011 C' = 01010111
K: 10100111 E(R1) = 0110 1001 K2 = 10100111 11001110 Sbox = 1110 f = 1101 L1 = 1010 R2 = 0111 L2R2 = 1100 0111 *C' = 01010111 C' = 01010111	K: 1111011 E(R1) = 0110 1001 K2 = 11110111 10011110 Sbox = 0110 f = 1001 L1 = 1010 R2 = 0011 L2R2 = 1100 0011 *C' = 01010011 C' = 01010111

Dari hasil perhitungan diatas didapat bahwa kunci 00000111 dan 10100111 yang digunakan untuk mengenkripsi menghasilkan *chipertext* yang sama dengan *chipertext* hasil enkripsi SDES. Sehingga didapatkan K2 = 00000111 dan 10100111. Perhitungan ini dilakukan untuk semua jumlah *plainteks* sehingga didapat nilai kesamaan pada C' yang berbeda dimana nilai kesamaan terbesar merupakan kunci yang benar. Kemudian hasil ini akan dibandingkan dengan K2 yang asli. Untuk DES 8 putaran proses pengujian dilakukan sama dengan pada SDES .

BAB IV
HASIL DAN PEMBAHASAN

Bab ini akan membahas hasil implementasi yang dihasilkan oleh perangkat lunak, untuk selanjutnya dilakukan evaluasi kunci yang didapatkan dari proses *differensial cryptanalisis.*

4.1 Lingkungan Implementasi

Lingkungan implementasi meliputi lingkungan perangkat keras serta lingkungan perangkat lunak.

4.1.1 Lingkungan Perangkat Keras

Perangkat keras yang digunakan dalam implementasi *differensial cryptanalisis* adalah:

1. Prosesor Intel(R) Pentium(R) 4 CPU 1.70GHz
2. RAM 384 MB
3. *Harddisk* dengan kapasitas 20 GB
4. Monitor
5. Keyboard
6. Mouse

4.1.2 Lingkungan Perangkat Lunak

Perangkat lunak yang digunakan dalam implementasi *differensial cryptanalisis* adalah:

1. Sistem Operasi *Microsoft Windows XP Professional*.
2. *Microsoft Visual* C# 2008 *Express Edition.*

4.2 Implementasi Program

Berdasarkan perancangan perangkat lunak pada subbab 3.2 maka pada subbab ini akan dibahas mengenai implementasi dari perancangan tersebut.

4.2.1 Input

U*ser* melakukan input yang berupa kunci, file yang akan di*enkripsi*, algoritma yang diinginkan. Untuk *differensial chyptanalisis* inputan berupa ciphertext pada putaran terakhir dan sebelum putaran terakhir.

Pada proses input file yang diambil diletakkan dalam string yang nanti digunakan sebagai masukan dalam proses *enkripsi.* Pada *class* biner terdapat fungsi binToDes, desTobin, gabungBiner, XOR, geserKiri.

```
public static int binToDes(bool[] binBool)
{
for (int i = 0; i < binBool.Length; i++)
{
 if (binBool[i] == true)
 { bin += "1"; }

 else
 { bin += "0"; }
 return Convert.ToInt16(bin, 2);
}

public static bool[] desTobin(int des)
{
 int pembagi = des;
 while (pembagi != 0)
 {
  if (pembagi % 2 == 0)
  { binerdibalik += "0"; }

  else
  { binerdibalik += "1"; }
   pembagi = ((pembagi - (pembagi % 2)) / 2);
 }

for (int i = 0; i < binerdibalik.Length; i++)
{
 biner += binerdibalik[binerdibalik.Length-i,1]);
}
bool[] jawab = new bool[biner.Length];

for (int i = 0; i < biner.Length; i++)
{
 if (biner[i] == '1')
 { jawab[i] = true; }
else
{ jawab[i] = false; }
 return jawab;}
```

***Sourcecode* 4.1** *Sourcecode* biner

Pada *Sourcecode* 4.1 terdapat fungsi untuk mengubah desimal menjadi biner dan sebaliknya biner menjadi desimal dimana digunakan untuk mengubah nilai ASCII dari inputan menjadi biner sehingga dapat dilakukan proses *enkripsi*, serta terdapat juga fungsi untuk menggabung dan menggeser bilangan yang telah diubah menjadi biner yang digunakan pada proses pembentukan kunci serta proses *enkripsi*.

```
public static ArrayList fusion(row, col)
{
for (int i = 0; i < row.Count; i++)
{
 for (int k = 0; k < tinggi; k++)
 {
  temp[0] = (row[i])[k])[0];
  temp[lebar-1]=(row[i])[k])[1];

  for (int j = 1; j <= lebar - 2; j++)
  {
   temp[j] = (col[i])[k])[j - 1];
   arrayTemp.Add(temp);
   jawab.Add(arrayTemp);
  }
```

***Sourcecode* 4.2** *Sourcecode* fungsi gabung

Sourcecode 4.2 berisi fungsi untuk menggabung baris dan kolom yang telah didapat dari S_box. Fungsi ini berada pada *class* MyUtil yaitu fungsi fusion.

4.2.2 Enkripsi

Proses untuk mengubah *plainteks* menjadi *cipherteks* dengan menggunakan kunci yang telah dimasukkan oleh *user* sehingga isi file atau masukkan tidak dapat diketahui oleh orang yang tidak berkepentingan. Hasil dari proses enkripsi ini akan disimpan 5 file yaitu file *plainteks,* file *plainteks* korespondensi, file *cipherteks,* file *cipherteks* korespondensi, dan kunci.

4.2.2.1 SDES

Proses *enkripsi* dengan kunci sepanjang 10 bit dan dilakukan dalam 2 putaran.

```
public SDES(char _kunci, String _plaintext)
{
  cariKunciHasil();
  EnkripPlainTeks();
}
```

***Sourcecode* 4.3** *Sourcecode* SDES

Pada proses *enkripsi* pada SDES terdapat fungsi untuk mendapatkan kunci (cariKunciHasil) serta untuk meng*enkripsi* file (EnkripsiPlainTeks).

```
 private void cariKunciHasil()
 {
  acakKunci();
  cariC0D0();
  cariC1D1();
  cariC2D2();
  bool[] gabung1 = Biner.gabungBiner(c1, d1);
  bool[] gabung2 = Biner.gabungBiner(c2, d2);
  k1 = acakDelapanBitKunci(gabung1);
  k2 = acakDelapanBitKunci(gabung2);
 }
```

***Sourcecode* 4.4** *Sourcecode* cari kunci pada SDES

Pada *Sourcecode* 4.4 pada fungsi cariKunciHasil terdapat fungsi acakKunci yang dimana kunci yang telah dimasukkan dan diubah menjadi 10 bit biner diacak dan kemudian dibagi menjadi 2 yaitu C0 dan D0 yang terdapat pada cariC0D0 kemudian untuk menghasilkan kunci pada putaran 1 (k1), C0 dan D0 digeser kiri 1 bit yang terdapat pada fungsi cariC1D1 dan juga untuk menghasilkan kunci putaran 2 (k2) maka C1 dan D1 digeser kiri 2 bit kemudian hasil pergeseran digabung kemudian diacak dan tersisa 8 bit kunci untuk putaran 1 (k1) dan putaran 2 (k2).

```
 private void EnkripPlainTeks()
  {
   ubahPlainTeksKeBiner();
```

```
    cariIp1();
    cariL0R0();
    cariL1();
    cariHasilXor1();
    cariR1L2();
    cariHasilXor2();
    chipertext();
  }
```

***Sourcecode* 4.5** *Sourcecode enkripsi* pada SDES

Pada *Sourcecode* 4.5 pada fungsi EnkripsiPlainTeks terdapat fungsi untuk mengubah *plainteks* (file) menjadi biner agar dapat di*enkripsi* (ubahPlainTeksKeBiner) kemudian hasilnya dimasukan pada Ip1 (cariIp1) kemudian dicari L0R0 dari *plainteks* yang telah dimasukkan ke tabel Ip1 (cariL0R0), pada fungsi cariL1 didapatkan L1 yang sama dengan R0. Setelah itu pada cariHasilXor1 dilakukan XOR antara L1 dengan k1, hasil XOR tersebut dimasukkan kedalam tabel S-Box yang kemudian dilakukan fungsi f yaitu mengacak hasil dari S-Box. Hasil keluaran fungsi f diXORkan dengan L0 untuk menghasilkan R1 (cariR1L2), untuk mendapatkan L2 dan R2 proses sama dengan mendapatkan L1 dan R1 hanya yang digunakan untuk mendapatkan L1, R1 dan k2. Setelah didapatkan L2R2, hasilnya dimasukkan kedalam IP Invers untuk mendapatkan *chiperteks* (pada fungsi chiperteks). Hasil dari proses ini adalah *chiperteks* dari file yang di*enkripsi*.

4.2.2.2 DES 8 Putaran

Proses *enkripsi* dengan kunci sepanjang 64 bit dan dilakukan dlam 8 putaran.

```
public DES8(String _kunci, String _plaintext)
{
 cariKunciHasil();
 EnkripPlainTeks();
}
```

***Sourcecode* 4.6** *Sourcecode* DES 8 putaran

Pada proses *enkripsi* pada DES 8 putaran terdapat fungsi untuk mendapatkan kunci (cariKunciHasil) serta untuk meng*enkripsi* file (EnkripsiPlainTeks) seperti pada SDES.

```
  private void cariKunciHasil()
```

```
 {
  acakKunci();
  cariC0D0();
  cariC1D1();
  cariC2D2();
  cariC3D3();
  cariC4D4();
  cariC5D5();
  cariC6D6();
  cariC7D7();
  cariC8D8();
 }
```

***Sourcecode* 4.7** *Sourcecode* cari kunci pada DES 8 putaran

Pada *Sourcecode* 4.7 pada fungsi cariKunciHasil terdapat fungsi acakKunci yang dimana kunci yang telah dimasukkan dan diubah menjadi 56 bit biner diacak dan kemudian dibagi menjadi 2 yaitu C0 dan D0 yang terdapat pada cariC0D0 kemudian untuk menghasilkan kunci pada putaran 1 (k1), C0 dan D0 digeser kiri 1 bit yang terdapat pada fungsi cariC1D1 dan juga untuk menghasilkan kunci putaran 2 (k2) maka C1 dan D1 digeser kiri 2 bit dan seterusnya sampai didapat C8 dan D8, kemudian hasil pergeseran digabung kemudian diacak dan tersisa 48 bit kunci untuk putaran 1 (k1) dan putaran 2 (k2) sampai putaran 8 (k8).

```
private void EnkripPlainTeks()
{
 ubahPlainTeksKeBiner();
 cariIp1();
 cariL0R0();
 cariL1();
 cariHasilXor1();
 cariR1L2();
 cariHasilXor2();
 cariR2L3();
 cariHasilXor3();
 cariR3L4();
 cariHasilXor4();
 cariR4L5();
 cariHasilXor5();
 cariR5L6();
 cariHasilXor6();
 cariR6L7();
```

```
 cariHasilXor7();
 cariR7L8();
 cariHasilXor8();
 chipertext();
}
```

***Sourcecode* 4.8** *Sourcecode enkripsi* pada DES 8 putaran

Pada *Sourcecode* 4.8 pada fungsi EnkripsiPlainTeks terdapat fungsi untuk mengubah *plainteks* (file) menjadi biner agar dapat di*enkripsi* (ubahPlainTeksKeBiner) kemudian hasilnya dimasukan pada Ip1 (cariIp1) kemudian dicari L0R0 dari *plainteks* yang telah dimasukkan ke tabel Ip1 (cariL0R0), pada fungsi cariL1 didapatkan L1 yang sama dengan R0. Setelah itu pada cariHasilXor1 dilakukan XOR antara L1 dengan k1, hasil XOR tersebut dimasukkan kedalam tabel S-Box yang kemudian dilakukan fungsi f yaitu mengacak hasil dari S-Box. Hasil keluaran fungsi f diXORkan dengan L0 untuk menghasilkan R1 (cariR1L2), untuk mendapatkan L2 dan R2 proses sama dengan mendapatkan L1 dan R1 hanya yang digunakan untuk mendapatkan L1, R1 dan k2 proses dilakukan sampai 8 putaran sampai mendapat L8 dan R8. Setelah didapatkan L8R8, hasilnya dimasukkan kedalam IP Invers untuk mendapatkan *chiperteks* (pada fungsi chiperteks). Hasil dari proses ini adalah *chiperteks* dari file yang di*enkripsi*.

4.2.3 *Differensial Cryptanalisis*

Proses *differensial cryptanalisis* untuk mendapatkan kemungkinan kunci pada putaran terakhir. Untuk *differensial cryptanalysis* digunakan 4 file hasil dari proses enkripsi yaitu file *plainteks,* file *pleinteks* korespondensi, file *cipherteks,* file *cipherteks* korespondensi.

4.2.3.1 SDES

Proses *differensial cryptanalisis* untuk mendapatkan kemungkinan kunci pada putaran terakhir (K2) sehingga dapat diperoleh 8 bit kunci dari 10 bit kunci asli.

```
public DifferentSDES(SDES _teks)
```

```
{
  carix0y0put2();
  carix1y1put2();
  ambil();
  mngkn_kunci();
}
```

***Sourcecode* 4.9** *Sourcecode differentsial* pada SDES

Pada *Sourcecode* 4.9 dicari delta x0, y0, x1, y1 pada putaran terakhir (putaran 2) untuk melakukan *differentsial cryptanalisis* (carix0y0put2 dan cari x1y1put2), kemudian diambil kemungkinan input untuk S-Box pada fungsi ambil, kemudian dicari kemungkinan kunci pada putaran terakhir menggunakan *differentsial cryptanalisis* pada fungsi mngkn_kunci.

```
 private void mngkn_kunci()
{
for (int k = 0; k < rowbin.Count - 1; k++)
 {
 for (int m = 0; m < 4; m++)
 {
  while (x < 4)
  {
   if ((k + 1) < rowbin.Count)
   { hsl =XOR(inputS0[k][m],inputS0[k+1])[x]); }

//untuk s0
   if ((simpanx0_2[k]) == hsl)
   { hslXORs0_1.Add(((inputS0[k])[m])); }
     x++; }

 for (int n = 0; n < hslXORs0_1.Count; n++)
 {
   hasil = XOR((hslXORs0_1[n]),ptextp2_0[k]);
   kuncis0.Add(hasil);
 }
```

***Sourcecode* 4.10** *Sourcecode* kemungkinan kunci pada SDES

Hasil dari proses ini adalah kemungkinan kunci pada putaran terakhir SDES (K2).

4.2.3.2 DES 8 Putaran

Proses *differensial cryptanalisis* untuk mendapatkan kemungkinan kunci pada putaran terakhir (K8) sehingga dapat diperoleh 48 bit kunci dari 56 bit kunci asli.

```
public differensial8ptrn(DES8 _teks)
 {
  carix1y1put8();
  carix2y2put8();
  carix3y3put8();
  carix4y4put8();
  carix5y5put8();
  carix6y6put8();
  carix7y7put8();
  carix8y8put8();
  ambil();
  mngkn_kunci();
 }
```

***Sourcecode* 4.11** *Sourcecode differentsial* pada DES 8 putaran

Pada *Sourcecode* 4.11 dicari delta x1, y1, sampai x8, y8 pada putaran terakhir (putaran 8) untuk melakukan *differentsial cryptanalisis* (carix1y1put8 sampai carix8y8put8), kemudian diambil kemungkinan input untuk S-Box pada fungsi ambil, kemudian dicari kemungkinan kunci pada putaran terakhir menggunakan *differentsial cryptanalisis* pada fungsi mngkn_kunci

```
 private void mngkn_kunci()
{
 for (int k = 0; k < rowbin1_8.Count - 1; k++)
 {
  for (int m = 0; m < 4; m++)
  {
   while (x < 4)
   {
   if ((k + 1) < rowbin1_8.Count)
    {hsl1 = XOR(inputS1[k])[m],inputS1[k + 1])[x]);}

//untuk s1
   if (simpanx1_8[k] == hsl1)
   {
    hslXORs1_8.Add(((inputS1[k])[m]));
    hslXORs1_8.Add(((inputS1[k + 1])[x]));
   }
   x++;}

 for (int n = 0; n < hslXORs1_8.Count; n++)
 {
  hasil1 = XOR(hslXORs1_8[n],ptextp8_1[k]);
  kuncis1.Add(hasil1);
 }
```

***Sourcecode* 4.12** *Sourcecode* kemungkinan kunci pada DES 8 putaran

Hasil dari proses ini adalah kemungkinan kunci pada putaran terakhir DES 8 putaran (K8).

4.2.4 Pengujian

Proses pengujian untuk mendapatkan kemungkinan kunci yang benar dari beberapa kemungkinan kunci yang didapat dari *differentsial cryptanalisis* sehingga dapat diperoleh kemungkinan kunci yang benar.

4.2.4.1 SDES

Proses pengujian untuk mendapatkan kemungkinan kunci yang benar dari beberapa kemungkinan kunci (K2) yang didapat dari *differentsial cryptanalisis* pada SDES sehingga dapat diperoleh kemungkinan kunci (K2) yang benar.

```
public ujiSDES(SDES _uji, DifferentSDES _kunci)
{
```

```
    pengujian();
}

private void pengujian()
{
 for (int i = 0; i < k2.Count; i++)
 {
  for (int j = 0; j < put2.Count; j++)
  {
   hasil = XOR(k2[i], put2[j]);
   s0 = cariDiTabelS0((rowKiri),(columnKiri));
   rowKanan = (tempKanan[0],tempKanan[3]};
   columnKanan =  {tempKanan[1],tempKanan[2]};
   s1 = cariDiTabelS1((rowKanan), (columnKanan));
   gabung = gabungBiner(s0_bool1, s1_bool1);

 if ((chiper) ==(c[j]))
 { count++; }

  key.Add(k2[i]);
   hitung.Add(count);
```

***Sourcecode* 4.13** *Sourcecode* Uji SDES

Hasil dari proses ini adalah kemungkinan kunci yang paling benar dari beberapa kemungkinan kunci yang didapat dari proses *differential cryptanalisis* beserta banyak iterasi dalam pengujian.

4.2.4.2 DES 8 Putaran

Proses pengujian untuk mendapatkan kemungkinan kunci yang benar dari beberapa kemungkinan kunci (K8) yang didapat dari *differentsial cryptanalisis* pada DES 8 putaran sehingga dapat diperoleh kemungkinan kunci (K8) yang benar.

```
public uji8putaran(DES8 _uji, differensial8ptrn _kunci)
  {
   pengujian();
  }

 private void pengujian()
  {
  for (int i = 0; i < k8.Count; i++)
  {
   for (int j = 0; j < put8.Count; j++)
```

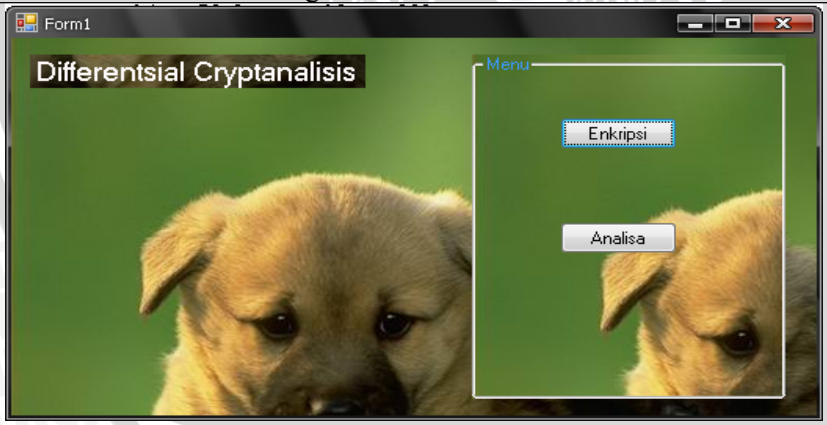
```
    {
     hasil = XOR(k8[i],put8[j]);

     row1 =  {temp1[0], temp1[5]};
     colum1 = {temp1[1],temp1[2],temp1[3],temp1[4]};

     s1 = cariDiTabelS1((row1),(colum1));

    gabung = gabungBiner(s1_bool1,s2_bool1);
    l8r8 = gabungBiner(l8[j], r8);

if    ((chiper1)==(asli1))&&((chiper2)==(asli2))
      &&((chiper3)==(asli3))&&((chiper4)==(asli4))
      &&((chiper5)==(asli5))&&((chiper6)==(asli6))
      && ((chiper7)==(asli7))&&((chiper8)==(asli8)))
     { count++; }

       key.Add(k8[i]);
       hitung.Add(count);
```

***Sourcecode* 4.14** *Sourcecode* Uji DES 8 putaran

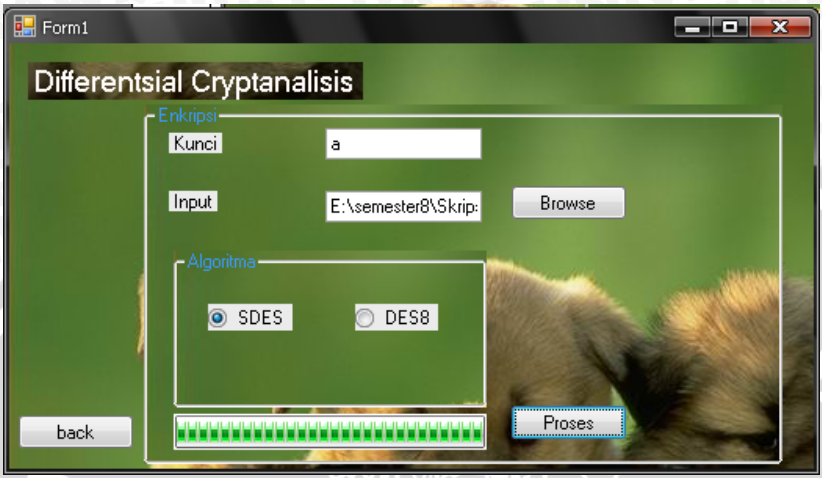
Hasil dari proses ini adalah kemungkinan kunci yang paling benar dari beberapa kemungkinan kunci (K8) yang didapat dari proses *differential cryptanalisis* beserta banyak pengujian.

4.3 Implementasi Antarmuka

Berdasarkan rancangan antarmuka pada bab sebelumnya, maka dibuatlah antarmuka sebagai berikut:

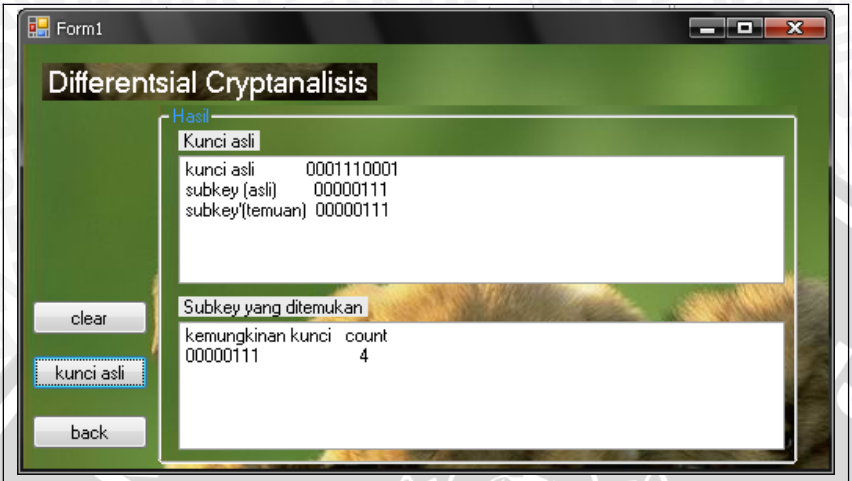


a. Form awal



b. Form Enkripsi

c. Form Analisa



d. Form Hasil

Gambar 4.1 Tampilan perangkat lunak

4.4 Implementasi Uji Coba

Pada subbab ini akan dilakukan pembahasan mengenai pengujian yang telah dilakukan pada sistem dan hasil evaluasi dari hasil yang dikeluarkan sistem. Pengujian dilakukan pada beberapa file yang berbeda ukuran dan panjang kata atau banyak huruf yang terdapat didalamnya dimana berpengaruh pada banyak pengujian. Tabel 4.1 merupakan file yang akan diuji pada sistem:

Tabel 4.1 File yang akan diuji

Nama file	Jumlah Blok	ukuran
harvestmoonislandofhappiness-walkthrough02.txt	19628	21 KB
harvestmoonislandofhappiness-walkthrough04.txt	22589	24 KB
Larangan di sekolah.doc	29124	29 KB
harvestmoonislandofhappiness-walkthrough01.txt	34746	37 KB
Doc2.doc	40957	41 KB
Harvest Moon.doc	163896	162 KB
Best Recipes vs.doc	273112	270 KB
harvestmoonislandofhappiness-walkthrough05.txt	256872	275 KB
Unlockable.doc	286264	283 KB
Rune_Factory_2_FAQ-Walkthrough.txt	303744	1099 KB

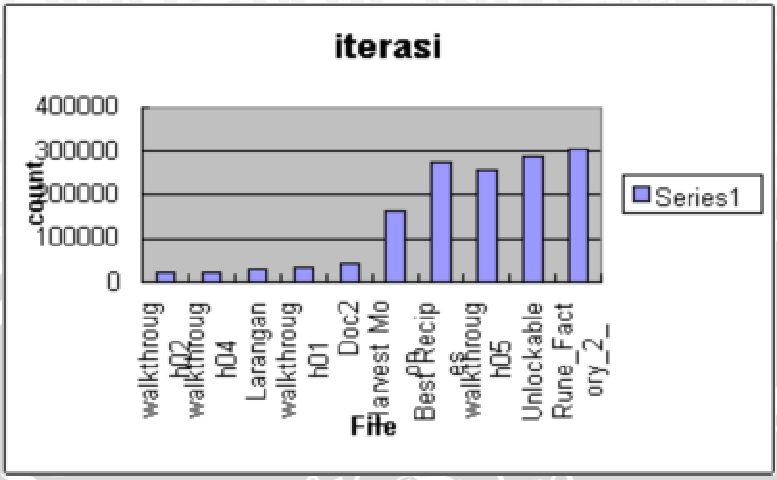
4.4.1 Evaluasi K2 pada SDES

Dilakukan pengujian terhadap 10 file dengan ukuran berbeda dimana dilakukan enkripsi dengan menggunakan SDES pada tiap file kemudian dilakukan *differensial cryptanalisis* dan terakhir dilakukan uji terhadap kunci putaran terakhir (K2) yang ditemukan. Hasil pengujian ditampilkan pada tabel 4.2 dan tabel 4.3

Tabel 4.2 Kemungkinan kunci pada SDES

File	Kunci	Count
harvestmoonislandofhappiness-walkthrough02.txt	00000111	19628
harvestmoonislandofhappiness-walkthrough04.txt	00000111	22589
Larangan di sekolah.doc	00000111	29124
harvestmoonislandofhappiness-walkthrough01.txt	00000111	34746
Doc2.doc	00000111	40957
Harvest Moon.doc	00000111	163896
Best Recipes vs.doc	00000111	273112
harvestmoonislandofhappiness-walkthrough05.txt	00000111	256872
Unlockable.doc	00000111	286264
Rune_Factory_2_FAQ-Walkthrough.txt	00000111	303744

Pada tabel 4.2 dapat dilihat perbedaan jumlah iterasi jika dilakukan pada file yang berbeda dimana semakin besar file yang digunakan maka jumlah iterasi semakin besar untuk pengujian dan mendapatkan kunci dimana kunci yang didapat harus benar untuk jumlah iterasi tersebut. Grafik 4.2 menunjukkan hasil banyak iterasi dari kemungkinan kunci (K2) yang didapat dari *differensial cryptanalisis* dimana pada tiap file, pengujian dilakukan menggunakan kunci yang sama.

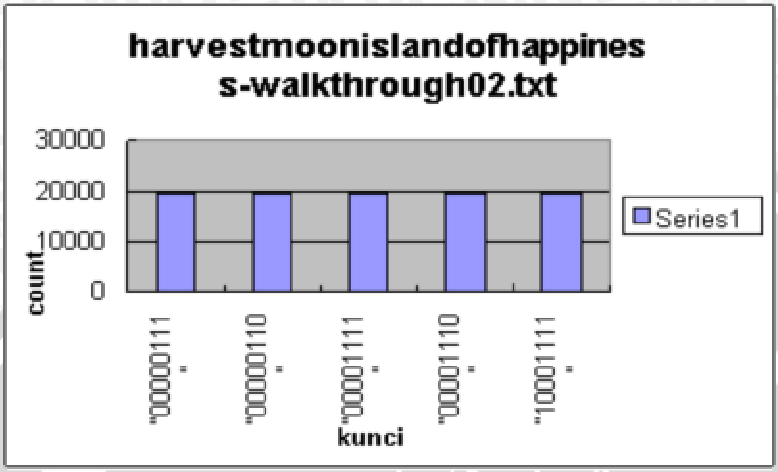


Gambar 4.2 Grafik jumlah iterasi dari file dengan kunci sama

Tabel 4.3 Kemungkinan kunci pada SDES dengan kunci berbeda

File	Kunci	Count
harvestmoonislandofhappiness-walkthrough02.txt	00000111	19628
harvestmoonislandofhappiness-walkthrough02.txt	00000110	19628
harvestmoonislandofhappiness-walkthrough02.txt	00001111	19628
harvestmoonislandofhappiness-walkthrough02.txt	00001110	19628
harvestmoonislandofhappiness-walkthrough02.txt	10001111	19628

Pada tabel 4.3 didapat hasil kunci dimana kunci berbeda tetapi dilakukan pengujian untuk file yang sama dimana jumlah iterasi yang didapat sama. Grafik 4.3 merupakan hasil kemungkinan kunci (K2) yang didapat dari *differensial cryptanalisis* dimana pada file, pengujian dilakukan menggunakan kunci yang berbeda.



Gambar 4.3 Grafik banyak iterasi yang dilakukan dari file harvestmoonislandofhappiness-walkthrough02.txt dengan kunci berbeda

4.4.2 Evaluasi K8 pada DES 8 Putaran

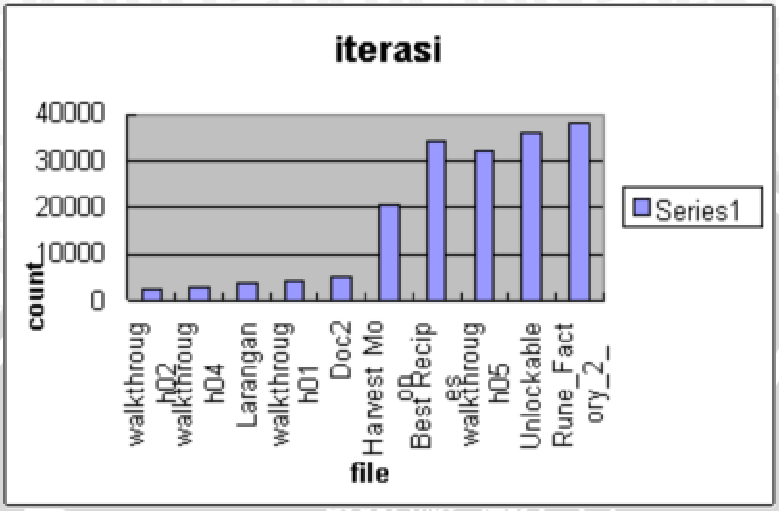
Dilakukan pengujian terhadap 10 file dengan ukuran berbeda dimana dilakukan enkripsi dengan menggunakan DES 8 putaran pada tiap file kemudian dilakukan *differensial cryptanalisis* dan terakhir dilakukan uji terhadap kunci putaran terakhir (K8) yang ditemukan. Hasil pengujian ditampilkan pada tabel 4.4 dan tabel 4.5

Tabel 4.4 Kemungkinan kunci pada DES 8 putaran

File	Kunci	Count
harvestmoonislandofhappines s-walkthrough02.txt	100111110101100111 011001010111000001 010100100110	2454
harvestmoonislandofhappines s-walkthrough04.txt	100111110101100111 011001010111000001 010100100110	2824
Larangan di sekolah.doc	100111110101100111 011001010111000001 010100100110	3641
harvestmoonislandofhappines	100111110101100111	4344

s-walkthrough01.txt	011001010111000001 010100100110	
Doc2.doc	100111110101100111 011001010111000001 010100100110	5120
Harvest Moon.doc	100111110101100111 011001010111000001 010100100110	20487
Best Recipes vs.doc	100111110101100111 011001010111000001 010100100110	34139
harvestmoonislandofhappines s-walkthrough05.txt	100111110101100111 011001010111000001 010100100110	32109
Unlockable.doc	100111110101100111 011001010111000001 010100100110	35783
Rune_Factory_2_FAQ-Walkthrough.txt	100111110101100111 011001010111000001 010100100110	37968

Pada tabel 4.4 dapat dilihat perbedaan jumlah iterasi jika dilakukan pada file yang berbeda dimana semakin besar file yang digunakan maka jumlah iterasi semakin besar untuk pengujian. Grafik 4.4 menunjukkan hasil kemungkinan kunci (K8) yang didapat dari *differensial cryptanalisis* dimana pada tiap file, pengujian dilakukan menggunakan kunci yang sama.

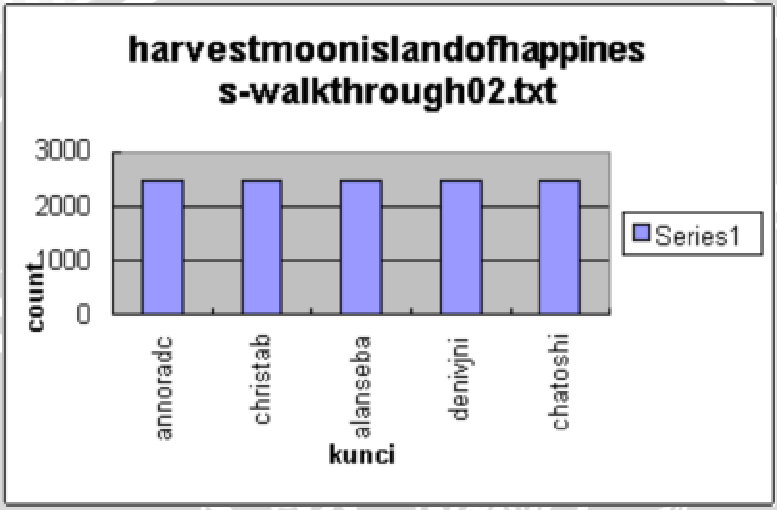


Gambar 4.4 Grafik jumlah iterasi dari file dengan kunci sama

Tabel 4.5 Kemungkinan kunci pada DES 8 putaran dengan kunci berbeda

File	Kunci	Count
harvestmoonislandofhappiness -walkthrough02.txt	10011111010110011 10110010101110000 01010100100110	2454
harvestmoonislandofhappiness -walkthrough02.txt	10011111010110011 10110110001000100 01000001101000	2454
harvestmoonislandofhappiness -walkthrough02.txt	10011111010110011 10110010001110000 00100001100010	2454
harvestmoonislandofhappiness -walkthrough02.txt	00011111010110111 10110010110001111 11000100000010	2454
harvestmoonislandofhappiness -walkthrough02.txt	00011111010110011 10110110011010001 00001010101001	2454

Pada tabel 4.5 didapat hasil kunci dimana kunci berbeda tetapi dilakukan pengujian untuk file yang sama dimana jumlah iterasi yang didapat sama. Grafik 4.5 menunjukkan hasil kemungkinan kunci (K8) yang didapat dari *differensial cryptanalisis* dimana pada file, pengujian dilakukan menggunakan kunci yang berbeda.



Gambar 4.5 Grafik banyak iterasi yang dilakukan dari file harvestmoonislandofhappiness-walkthrough02.txt dengan kunci berbeda

4.4.3 Analisa Hasil

Pada tugas akhir ini pengujian yang dilakukan dengan menggunakan algoritma SDES dan DES 8 putaran untuk melakukan enkripsi dan digunakan *differensial cryptanalisis* untuk mendapatkan kunci pada putaran terakhir serta pembahasan didasarkan pada banyaknya iterasi bukan berdasarkan waktu karena waktu tergantung pada efisiensi dari suatu program yang dibuat. File masukan dienkripsi terlebih dahulu kemudian dilakukan serangan *differensial cryptanalisis* untuk memperoleh kunci pada putaran terakhir dimana pada pengujian didapat bahwa *differensial cryptanalisis* yang dilakukan pada SDES dan DES 8 putaran dapat memperoleh kunci pada putaran terakhir (K2 pada SDES dan K8 pada DES 8 putaran). Kunci pada putaran terakhir tersebut merupakan bagian dari kunci asli dimana berarti didapat 8 bit

dari 10 bit kunci asli pada SDES serta 48 bit dari 64 bit kunci asli. Pada tabel 4.6 dan 4.7 dapat dilihat kunci asli, subkey, subkey yang didapat pada percobaan:

Tabel 4.6 Hasil uji *differensial cryptanalisis* pada SDES

Kunci	Subkey	Subkey'	nilai
0001100001 (a)	00000111	00000111	19628
0001110010 (r)	00001110	00001110	22589
0001100111 (g)	10001111	10001111	29124

Tabel 4.6 menunjukkan hasil pengujian dari *diffrensial cryptanalisis* pada SDES dimana didapatkan *subkey'* atau kunci pada putaran terakhir sama dengan *subkey* yang asli dimana percobaan dilakukan dengan kunci yang berbeda dan file yang berbeda yang menghasilkan jumlah iterasi yang berbeda untuk file yang berbeda dimana jumlah iterasi tergantung pada besarnya file.

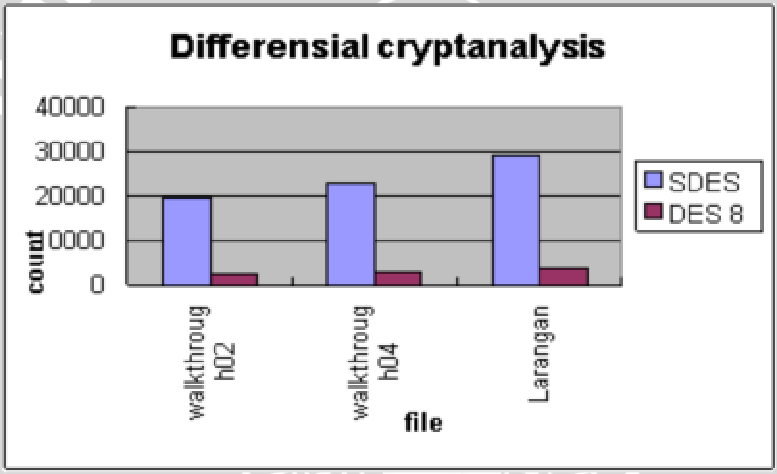
Tabel 4.7 Hasil uji *differentsial cryptanalisis* pada DES 8 putaran

Kunci	Subkey	Subkey'	nilai
0110000011011 1011011101101 1101110010110 0000110010011 0001 (annoradc)	10011111010 11001110110 01010111000 00101010010 0110	100111110 101100111 011001010 111000001 010100100 110	2454
0110001011010 0011100101101 0001110010111 0100110000011 0001 (christab)	10011111010 11001110110 11000100010 00100000110 1000	100111110 101100111 011011000 100010001 000001101 000	2824
0110000011011 0011000001101 1101110010110 0100110001011	10011111010 11001110110 01000111000 00010000110	100111110 101100111 011001000 111000000	3641

0000 (alanseba)	0010	100001100 010	

Tabel 4.7 menunjukkan hasil pengujian dari *diffrensial cryptanalisis* pada DES 8 putaran dimana didapatkan *subkey'* atau kunci pada putaran terakhir sama dengan *subkey* yang asli dimana percobaan dilakukan dengan kunci yang berbeda dan file yang berbeda yang menghasilkan jumlah iterasi yang berbeda untuk file yang berbeda dimana jumlah iterasi tergantung pada besarnya file.

Grafik 4.6 diperoleh dari tabel 4.6 dan 4.7 dimana menunjukkan pengujian dilakukan pada SDES dan DES 8 putaran dengan file yang sama.



Gambar 4.6 Grafik *different cryptanalisis* pada SDES dan DES 8 putaran diuji pada file yang sama

SDES mengubah tiap karakter pada file (1 karakter diubah menjadi 8 bit) untuk melakukan *enkripsi, differensial cryptanalisis,* serta pengujian sedangkan DES 8 putaran mengubah tiap 8 karakter pada file (8 karakter diubah menjadi 64 bit) untuk melakukan *enkripsi, differensial cryptanalisis,* serta pengujian oleh karena itu pada file yang sama jumlah iterasi untuk SDES dan DES 8 putaran berbeda dimana jumlah iterasi SDES lebih besar 8 kali dan lebih lama daripada DES 8 putaran .


UNIVERSITAS BRAWIJAYA

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Kesimpulan yang diperoleh selama pengerjaan Skripsi ini adalah:

1. *Subkey* pada putaran terakhir didapat dengan mencari *difference* dari p*lainteks* dan *cipherteks*nya kemudian didapat *subkey* dengan count tertinggi merupakan *subkey* yang benar.
2. *Differensial cryptanalysis* terhadap SDES dan DES 8 putaran menghasilkan jumlah *subkey* cocok sebanyak 8 bit untuk SDES dan 48 bit untuk DES 8 putaran.

5.2 Saran

Saran untuk pengembangan lebih lanjut yang dapat diberikan oleh penulis adalah:

1. Mencari metode dan mengimplementasikannya untuk menemukan sisa kunci yang tidak ditemukan dengan metode *differensial cryptanalysis*.


UNIVERSITAS BRAWIJAYA

DAFTAR PUSTAKA

Ariyus, Dony. 2008. Pengantar Ilmu Kriptografi. Yogyakarta:Andi

Chin, Brain. 2002. Crptanalysis of S-DES. University of Sheffield Center:Taylor College

Heys, Howard. A Tutorial on Linear and Differential Cryptanalysis.Electrical and Computer Engineering Faculty of Engineering and Applied Science.University of Newfoundland. Canada

Setiadi, Budi. 2004. Analisis Sistem Keamanan Data dengan Menggunakan Metode DES dan Metode Ghost. Teknik Elektro Institut Teknologi Bandung

Sipahutar, Muara. 2004. Berbagai Kasus Penyerangan terhadap Kriptografi. Teknik Informatika Institut Teknologi Bandung http://www.informatika.org/~rinaldi/Kriptografi/Makalah/Makalah1-060.pdf.
Diakses tanggal 10 Januari 2009.

Biham, E. and A. Shamir. 1992. Differential Cryptanalysis of the Full 16-round DES. *Advances in Cryptology -- CRYPTO '92*. Springer-Verlag.

Miles E. Smid and Dennis K. Branstad, "The Data Encryption Standard: Past and Future," in Gustavus J. Simmons, ed., *Contemporary Cryptography: The Science of Information Integrity*, IEEE Press, 1992.
http://home.earthlink.net/~mylnir/crypt.notes.
Diakses tanggal 15 Oktober 2008.

Carl H. Meyer and Stephen M. Matyas, *Cryptography: A New Dimension in Computer Data Security*, John Wiley & Sons, New York, 1982.
http://nsfsecurity.pr.erau.edu/crypto.
Diakses tanggal 10 Januari 2009.

UNIVERSITAS BRAWIJAYA

Lampiran 1: coba.txt (DES 8 putaran)

Plaintext:
halo apa kabar?

Cipher:
101111001100010011110110100010110110110110011011011010001011010100110111010101011001101010011010111001011101101110101101001000011

Kunci:
0110000011011101101110110111001000000100000010000001000000

Subkey:
000011110101000101011001000111000001010100000110


UNIVERSITAS BRAWIJAYA

Lampiran 2: Difference Distribution Table

S0:

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

S1:

ΔX	ΔY			
	0	1	2	3
0	16	0	0	0
1	2	6	2	6
2	0	8	0	8
3	6	2	6	2
4	0	8	4	4
5	6	2	6	2
6	0	0	12	4
7	2	6	2	6
8	2	6	2	6
9	0	0	12	4

10	6	2	6	2
11	0	8	4	4
12	6	2	6	2
13	4	4	0	8
14	2	6	2	6
15	12	4	0	0