Desperate Block Cipher

Lukas Kurnia Jonathan[1], **Rika Dewi**[2].

[1,2] Program Studi Teknik Informatika, Sekolah Teknik Elektro dan Informatika (STEI), Institut Teknologi Bandung (ITB), Jalan Ganesha 10, Bandung 40132
E-mail: 13517006@std.stei.itb.ac.id, 13517147@std.stei.itb.ac.id

Abstrak. Kriptografi modern memiliki gagasan yang serupa dengan kriptografi klasik, tetapi lebih kompleks karena menggunakan operasi bit dalam penerapannya. Desperate Block Cipher dibuat dengan menerapkan prinsip pada algoritma kriptografi modern menggunakan cipher blok dengan operasi ECB, CBC, dan Counter sebagai mode utama dalam pengoperasiannya. Dalam pengembangannya, Desperate Block Cipher menerapkan prinsip *confusion* dan *diffusion*, serta mengimplementasikan cipher putaran menggunakan jaringan Feistel untuk meningkatkan keamanan dari pesan. Berdasarkan hasil eksperimen dan analisis yang dilakukan terhadap Desperate Block Cipher menggunakan analisis *key space*, statistik, dan sensitivitas perubahan, algoritma ini memiliki tingkat keamanan yang bagus dan mampu mengatasi serangan *brute force* dan analisis statistik. **Kata kunci:** kriptografi modern, Desperate, cipher blok, *confusion*, *diffusion*, Feistel.

1. Pendahuluan

Pada era modern ini teknologi informasi semakin hari semakin berkembang. Ditambah dengan situasi pada tahun 2020 yang mendorong aktivitas dilakukan tanpa tatap muka membuat pengguna internet menjadi semakin banyak. Kebutuhan seseorang untuk berkomunikasi tidak lagi dilakukan secara langsung, tetapi menggunakan media digital untuk mengirimkan suatu pesan. Salah satu hal yang menjadi perhatian adalah aspek keamanan informasi ketika suatu pesan akan dikirimkan. Akan tetapi, dengan pengetahuan dan teknologi yang ada, penyadap dapat dengan mudah mengambil pesan yang sedang ditransmisikan[2].

Salah satu cara yang digunakan untuk menjaga aspek keamanan dari pesan yang dikirimkan adalah menggunakan kriptografi. Kriptografi merupakan suatu ilmu yang digunakan untuk menjaga keamanan pesan dengan menerapkan teknik-teknik matematika pada pesan yang dikirimkan[1]. Ilmu kriptografi yang pada mulanya beroperasi dalam mode karakter, semakin hari semakin berkembang menjadi kriptografi modern yang menerapkan operasi bit atau *byte* yang lebih kompleks untuk meningkatkan keamanan pesan. Meskipun dalam kriptografi modern menggunakan komputer digital untuk merepresentasikan data dalam biner, gagasan utama pada kriptografi klasik seperti transposisi dan subtitusi tetap digunakan[1].

Cipher blok merupakan salah satu jenis kriptografi modern yang beroperasi dalam bit. Beberapa diantaranya yaitu DES, Triple DES, AES dan cipher blok lainnya. Kebanyakan cipher blok beroperasi pada ukuran blok tertentu dengan menggunakan panjang kunci yang sudah didefinisikan. Berbagai operasi seperti putaran, permutasi, pertukaran juga dilakukan dalam menerapkan algoritma yang ada [1]. Sebagai contoh, DES beroperasi pada ukuran blok 64 (enam puluh empat) bit dengan panjang kunci

eksternal sebesar 64 (enam puluh empat) bit untuk membangkitkan 48 (empat puluh delapan) bit kunci internal. Algoritma DES sendiri melakukan enkripsi sebanyak 16 (enam belas) putaran[3].

Desperate Cipher Block merupakan block cipher yang dibuat karena terinspirasi dari algoritma DES, namun dengan meningkatkan jumlah kunci, blok yang digunakan, jumlah putaran, cara pembentukan kunci, dan prinsip-prinsip dalam cipher block. Penjelasan lebih lanjut akan dibahas pada jurnal ini dengan susunan sebagai berikut. Bagian dua akan membahas hasil studi pustaka yang relevan dengan topik yang diangkat. Bagian tiga membahas tentang rancangan detail dari Desperate Block Cipher. Bagian keempat akan menjelaskan hasil eksperimen dan analisis yang dilakukan. Kemudian ditutup oleh kesimpulan dan saran pada bab kelima.

2. Studi Literatur

Pada bagian ini akan dijelaskan hasil studi literatur yang berhubungan dengan cipher blok dan teori pendukung lain.

2.1. Cipher Block

Cipher blok merupakan salah satu jenis kriptografi modern selain cipher alir yang beroperasi dalam bit. Pada cipher blok, pesan dibagi menjadi blok-blok bit dengan panjang yang sama, misalkan 64 bit, 128 bit, dan 256 bit. Blok cipherteks hasil akan memiliki panjang yang sama dengan panjang blok plainteks, sedangkan untuk kunci eksternal bisa saja memiliki panjang yang berbeda dengan blok plainteks[1].

Blok plainteks berukuran n bit:

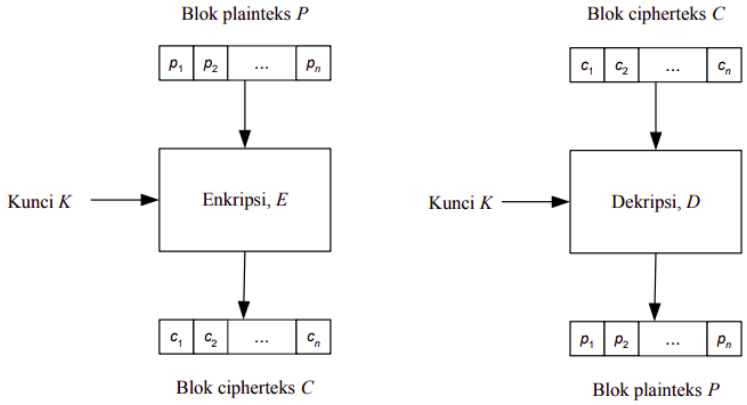
$P = (p_1, p_2, ..., p_n), \quad p_i \in \{0, 1\}$

Blok cipherteks (C) berukuran n bit:

$C = (c_1, c_2, ..., c_n), \quad c_i \in \{0, 1\}$

Gambar 2.1. Blok cipher [1]

Enkripsi dan dekripsi pada cipher blok dilakukan pada masing-masing blok seperti yang dapat dilihat pada Gambar 2.2.



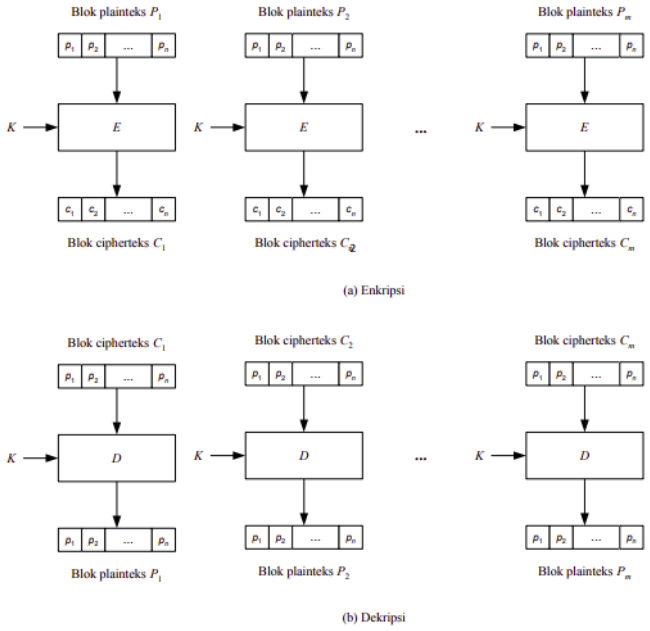
Gambar 2.2. Skema enkripsi dan dekripsi pada cipher blok [1]

2.2. Mode Operasi Cipher Blok

Pada bagian ini akan dijelaskan bagaimana sebuah blok dioperasikan sebelum dilakukan tahapan enkripsi dan dekripsi menggunakan fungsi enkripsi dan dekripsi. Mode operasi yang digunakan pada Desperate Block Cipher adalah *Electronic Code Book* (*ECB*), *Cipher Block Chaining* (*CBC*), dan *Counter Mode*.

2.2.1. Electronic Code Book (*ECB*)

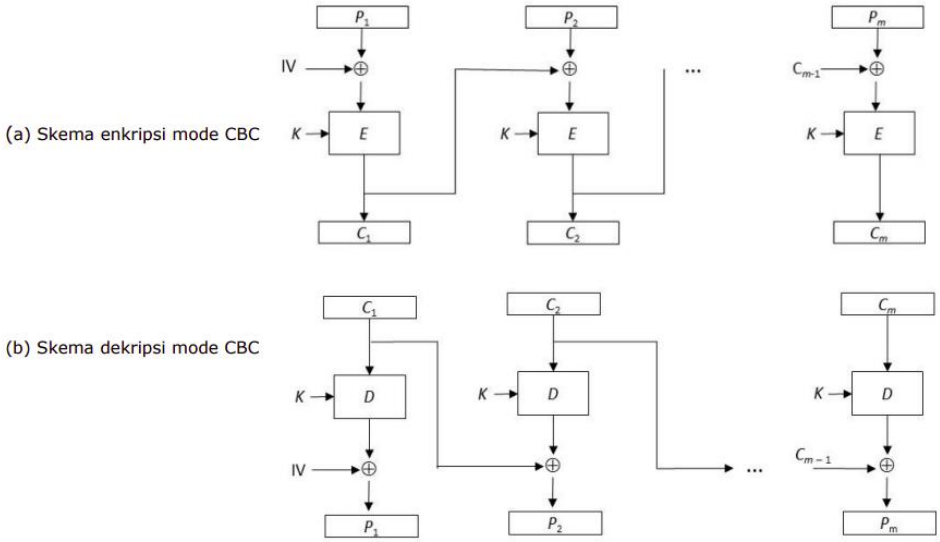
ECB (*Electronic Code Book*) merupakan salah satu mode operasi yang paling sederhana dengan mengenkripsi masing-masing blok dengan sebuah blok kunci. Enkripsi pada satu blok dilakukan secara independen dengan blok lainnya sehingga enkripsi setiap blok dapat dilakukan secara acak. Akan tetapi kelemahan dari mode operasi ini adalah banyaknya blok cipherteks yang sama dikarenakan banyak bagian berulang[1].



Gambar 2.3. Mode operasi ECB [1]

2.2.2. Cipher Block Chaining (*CBC*)

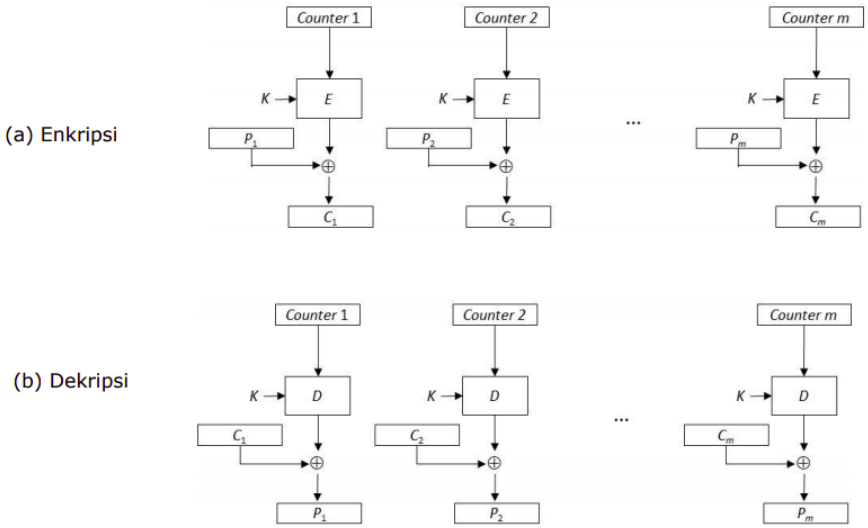
Dikarenakan kelemahan yang dimiliki oleh EBC, mode operasi CBC menggunakan ketergantungan antar blok dalam operasi yang dilakukannya. Masing-masing blok cipherteks tidak hanya bergantung kepada blok plainteks yang bersesuaian, tetapi juga terhadap blok plainteks sebelumnya. Enkripsi blok pertama dilakukan dengan melakukan operasi XOR kepada sebuah blok semu (C_0) atau IV (*initialization vector*) yang dapat didefinisikan pengguna atau dibangkitkan secara acak. Pada dekripsi, blok plainteks didapatkan dengan melakukan operasi XOR hasil dekripsi blok cipherteks terhadap IV[1].



Gambar 2.4. Mode operasi CBC [1]

2.2.3. Counter Mode

Berbeda dengan CBC, counter mode tidak melakukan proses *chaining*. Counter adalah blok bit yang memiliki ukuran sama dengan blok plainteks. Nilai counter untuk setiap blok akan berbeda satu dengan lainnya. Misalkan pada blok pertama nilai counter diinisialisasi dengan sebuah nilai, maka blok berikutnya nilai counter akan dinaikkan sebanyak satu[1].



Gambar 2.5. Mode operasi Counter [1]

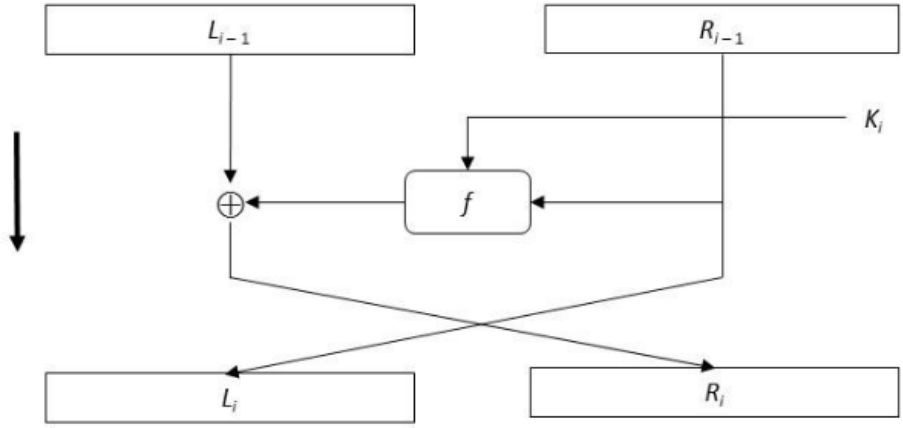
2.3. Aspek Keamanan Cipher Block

Hasil dari enkripsi sebuah pesan dengan menggunakan cipher blok tentunya harus mempertimbangkan aspek keamanan. Salah satu cara untuk meningkatkan keamanan dari cipher blok adalah menggunakan pembangkit permutasi pada kunci ketika proses melakukan enkripsi[4]. Kunci yang semakin sulit ditebak akan meningkatkan keamanan dari cipher blok sehingga hanya memberikan pilihan kepada kriptanalis untuk melakukan *exhastive key-search attack*[5]. Beberapa cara yang dapat dilakukan untuk

membuat kunci semakin sulit ditebak adalah menggunakan prinsip *Confusion* dan *Diffusion* yang akan dijelaskan pada sub bab selanjutnya.

2.4. Jaringan Feistel

Jaringan Feistel adalah sebuah skema yang membentuk komponen invertible (dapat dibalik) dengan menerapkan mekanisme *iterated cipher* (cipher berulang). Skema jaringan Fesitel ini juga memungkinkan penggunanya untuk melakukan enkripsi dan dekripsi dengan fungsi transformasi yang sama. Skema dari jaringan Fesitel ini dapat dilihat pada Gambar 2.6.



Gambar 2.6. Jaringan Feistel [1]

2.5. Diffusion dan Confusion

Prinsip *Diffusion* dan *Confusion* pertama kali diperkenalkan oleh Claude E. Shannon pada tahun 1949. Kedua prinsip ini bertujuan untuk mengatasi serangan yang bersifat statistik. *Confusion* menyatakan bahwa hubungan antara plainteks, cipherteks, dan kunci haruslah tersembunyi sehingga tidak mudah untuk ditemukan analisis korelasi hubungannya. *Diffusion* adalah prinsip yang membuat pengaruh perubahan 1 bit pada plainteks maupun kunci menjadi sangat besar pada cipherteks yang terbentuk.

3. Proposed Block Cipher

3.1. Gambaran umum

Desperate Cipher Block merupakan algoritma cipher blok yang terinspirasi dari cara kerja algoritma cipher blok DES (*Data Encryption Standard*) dengan jumlah variabel yang berbeda dan meningkatkan aspek keamanan. Algoritma Desperate ini dibuat dengan menggunakan bahasa pemrograman Python 3 untuk melakukan enkripsi dan dekripsi pesan yang beroperasi menggunakan bit.

3.1.1. Proses enkripsi

Proses enkripsi pada algoritma Desperate Block Cipher dilakukan dengan mengubah plainteks menjadi blok plainteks dengan panjang blok 128 bit. Pemrosesan akan blok sebelum dilakukan enkripsi akan disesuaikan dengan mode operasi yang dipilih seperti ECB, CBC atau Counter. Kemudian dilakukan permutasi plainteks berdasarkan *random seed* dari key. Selanjutnya, blok plainteks yang dihasilkan akan dimasukkan ke dalam jaringan Feistel yang sudah didefinisikan. Sebelum dilakukan proses enkripsi, kunci eksternal yang dimasukkan akan dibangkitkan menjadi 32 upa-kunci berbeda untuk setiap iterasi. Mengenai upa-kunci akan dijelaskan lebih lanjut pada Bab 3.4. Pada jaringan Feistel, blok plainteks akan dibagi dua menjadi blok kiri dan kanan kemudian dilakukan operasi XOR seperti yang

telah dijelaskan pada Bab 2. Hasil dari jaringan feistel yang berupa blok bit akan dikembalikan menjadi pesan cipherteks.

3.1.2. Proses dekripsi

Proses dekripsi pada algoritma Desperate Block Cipher memiliki alur pengerjaan yang serupa dengan proses enkripsi pada Bab 3.1.1. Perbedaanya pada proses dekripsi adalah mengubah cipherteks menjadi plainteks semula dengan masukkan kunci yang sama seperti melakukan proses enkripsi. Hal ini dapat dilakukan sesuai dengan teori dari jaringan Feistel yang dijelaskan pada Bab 2. Setalah keluar dari jaringan Feistel, dilakukan proses permutasi untuk mengembalikan posisi bit pada plainteks.

3.2. Variabel

Pada Desperate Block Cipher terdapat beberapa variabel yang digunakan untuk mendefinisikan parameter yang digunakan untuk melakukan enkripsi dan dekripsi, yaitu sebagai berikut.

a. Ukuran blok plainteks yang digunakan sebesar 128 (seratus dua puluh delapan) bit.

b. Panjang kunci eksternal sebesar 128 (seratus dua puluh delapan) bit dengan panjang kunci internal yang menjadi upa-kunci sebesar 80 (delapan puluh) bit.

c. Jumlah putaran yang digunakan pada jaringan Feistel sebanyak 32 (tiga puluh dua) putaran dengan setiap putaran menggunakan upa-kunci yang berbeda.

3.2.1. Initialization Vector (IV)

Pada mode operasi CBC, cipher block Desperate menggunakan 128-bit sebagai IV seperti ditunjukkan pada Gambar 3.1.

```
1   1, 0, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 0, 0, 0, 1,
2   0, 1, 1, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0,
3   0, 0, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 0,
4   0, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1, 1, 1, 0, 1,
5   0, 1, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 1, 1, 1,
6   0, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1,
7   1, 0, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 0,
8   0, 0, 1, 0, 0, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1,
```

Gambar 3.1. *Initialization Vector*

3.2.2. Counter Initialization

Pada mode operasi *counter mode*, digunakan 0 sebagai counter untuk cipher block pertama pada Desperate.

3.2.3. Kotak-S

Cipher block Desperate menggunakan 16 kotak-S berukuran 4x8 yang memetakan 5-bit menjadi 4-bit. Berikut 16 kotak-S yang digunakan pada block cipher Desperate.

```
============ s0 ============
12  5   6   13  14  13  11  8
8   0   8   11  15  13  14  14
8   9   7   8   3   15  0   4
5   6   15  13  4   0   13  2

============ s1 ============
13  4   3   12  5   11  4   9
14  14  15  12  4   9   10  12
10  12  3   6   11  12  8   9
4   9   11  13  1   11  4   4

============ s2 ============
6   5   2   8   7   15  4   3
5   15  14  3   2   12  12  14
10  11  6   3   2   12  7   9
5   1   12  8   5   7   3   9

============ s3 ============
11  15  3   11  2   2   11  14
9   10  14  5   13  13  12  2
4   8   15  7   15  15  11  12
12  6   11  5   3   8   14  8

============ s4 ============
9   11  15  13  4   10  7   15
7   5   0   9   4   1   14  5
5   15  11  0   14  12  3   12
4   5   0   15  10  6   15  4

============ s5 ============
12  13  5   1   2   1   13  5
2   8   11  14  6   7   9   8
4   10  14  5   6   9   12  8
15  6   10  15  4   9   0   13

============ s6 ============
8   15  11  1   3   0   5   12
10  9   13  3   14  8   0   14
6   4   12  11  4   10  2   12
9   6   5   12  15  4   12  8

============ s7 ============
2   0   6   13  7   13  15  13
11  3   5   5   8   8   9   3
7   15  5   2   9   12  10  10
4   14  5   0   9   6   4   12

============ s8 ============
15  7   3   5   13  0   6   3
0   1   4   3   12  2   7   11
13  6   13  13  11  9   4   9
8   5   14  4   0   11  11  7

============ s9 ============
12  5   7   8   15  13  2   12
0   9   12  8   10  4   15  0
1   4   5   6   11  4   7   1
0   7   5   6   9   6   0   1

============ s10 ============
11  7   12  11  5   12  0   15
14  8   2   2   11  10  3   13
8   5   2   13  13  7   15  9
12  2   1   9   3   3   3   14

============ s11 ============
14  1   12  1   15  2   2   7
1   8   4   1   5   3   5   1
4   12  0   6   4   1   9   3
1   11  15  8   2   8   8   11

============ s12 ============
13  8   13  2   13  12  4   0
9   7   8   13  5   6   5   13
6   4   11  14  14  7   9   5
2   9   15  11  13  10  5   6

============ s13 ============
0   14  11  3   9   13  1   7
10  11  0   0   9   8   2   6
9   1   15  3   11  13  9   6
0   5   1   4   9   7   13  10

============ s14 ============
1   10  10  6   2   4   8   3
4   5   1   10  2   3   2   12
11  4   8   12  13  1   6   9
2   1   2   0   10  2   0   3

============ s15 ============
6   14  9   9   4   2   14  10
13  4   8   2   14  12  6   9
5   8   0   4   15  13  6   1
6   3   6   1   11  0   12  9
```

Gambar 3.2. Kotak-S pada block cipher Desperate

3.2.4. Permutation Table

Tabel permutasi digunakan ketika membangun upa-kunci (K_n) untuk melakukan permutasi dari 96 (sembilan puluh enam) bit (K_x) menjadi 80 (delapan puluh) bit.

```
PB = [
    86, 5, 28, 74, 19, 9, 22, 3, 92, 66,
    36, 54, 33, 42, 35, 31, 47, 60, 4, 46,
    95, 71, 61, 37, 68, 24, 14, 90, 63, 52,
    39, 16, 15, 62, 2, 40, 67, 83, 58, 65,
    79, 72, 89, 8, 41, 23, 69, 38, 93, 32,
    48, 49, 85, 77, 45, 53, 20, 82, 57, 84,
    30, 59, 75, 18, 50, 21, 34, 1, 7, 43,
    70, 26, 44, 27, 6, 94, 25, 55, 78, 13
]
```

Gambar 3.3. *Permutation Table*

Sebagai contoh, bit ke-1 pada K_n adalah bit ke-86 pada K_x dan seterusnya. Demikian juga bit terakhir pada K_n adalah bit ke-13 pada K_x

3.2.5. Shift Table

Shift Table digunakan untuk mengacak posisi bit pada saat pembangunan upa-kunci dengan cara menggeser (*shift*) bit sebanyak jumlah tertentu ke kiri. Bit yang berada paling depan akan berputar menjadi bit paling belakang dan seterusnya. Upa kunci ke-n akan digeser mulai dari upa-kunci ke-(n-1)

```
SHIFT_TABLE = [
    1,2,1,2,1,3,2,1,
    3,2,1,2,1,1,2,3,
    2,1,1,2,2,2,2,2,
    2,1,3,1,1,1,1,2
]
```

Gambar 3.4. *Shift Table*

Sebagai contoh, upa-kunci ke-1 akan dilakukan pergeseran sebesar 1 kiri, upa kunci ke-32 akan dilakukan pergeseran sebesar 2 kali dari posisi awal upa kunci ke-31.

3.3. Permutasi Awal dan Akhir

Pada proses enkripsi, setiap blok pesan 128-bit dipermutasi terlebih dahulu sebelum masuk ke jaringan feistel. Permutasi ini dilakukan berdasarkan masukan kunci 128-bit. Pada proses dekripsi, blok pesan dikembalikan ke posisi semula setelah melewati jaringan feistel. Berikut adalah algoritma pembangkitan permutasi.

Mula-mula fungsi permutasi menerima masukkan sebuah *seed* untuk membangkitkan permutasi secara acak. *Seed* kunci yang dimasukkan akan dihitung terlebih dahulu nilai penjumlahan dari representasi ordinal setiap karakter dalam kunci. Sebagai contoh:

Kunci yang dimasukkan = “AAB”

Maka nilai pembangkit permutasi acak yang digunakan adalah seed = ord(A) + ord(A) + ord(B) = 65 + 65 + 66 = 196. Kemudian dilakukan operasi modulo antara seed dengan panjang dari target permutasi. Permutasi dilakukan dengan mengacak posisi dari setiap bit sesuai kelipatan dari seed yang digunakan. Sebagai contoh:

Target = ABCDEFG

Seed = 3

Maka hasil permutasi akan dimulai dari posisi A kemudian berjarak 3 (tiga) ke D, lalu G dan seterusnya. Sehingga hasil akhir dari permutasi adalah Target’ = ADGCFBE. Jika terdapat kasus dimana posisi dari huruf yang terpilih sudah digunakan, maka akan digunakan posisi huruf selanjutnya.

3.4. Fungsi transformasi

Bagian ini menjelaskan tentang fungsi tranformasi yang digunakan pada setiap iterasi cipher berulang. Fungsi ini menerima kunci 80-bit dan pesan 64-bit untuk menghasilkan string sepanjang 64-bit.

3.4.1. Substitusi

Fungsi ini menerima 80-bit input kunci dan mensubtitusinya menjadi output 64-bit dengan memanfaatkan matriks kotak-S. Sebuah kunci 80-bit, K_i akan dibagi menjadi menjadi 16 grup B_j dengan panjang 5-bit seperti ditunjukkan pada Gambar 3.5.

$$K_i = B_0 B_1 B_2 B_3 B_4 B_5 B_6 B_7 B_8 B_9 B_{10} B_{11} B_{12} B_{14} B_{15}$$

Gambar 3.5. Input kunci 80-bit

Setiap grup B_j akan disubsitusi menggunakan matriks kotak-S S_j. Proses substitusinya adalah sebagai berikut. Hasil desimal bit pertama dan kedua dari B_j menyatakan nomor baris dari matriks S_j, sedangkan hasil desimal bit ketiga, keempat, dan kelima secara berurutan menyatakan nomor kolom dari matriks S_j. Keluaran substitusi kotak-S yang berukuran 4-bit dari setiap dari setiap grup B_j akan digabungkan secara berurutan menghasilkan 64-bit kunci baru.

3.4.2. XOR

Pesan dengan penjang 64-bit dan 64-bit kunci baru kemudian dikenai operasi XOR secara *bitwise*.

3.5. Pembangkitan upa-kunci

Proses pembangkitan upa-kunci dilakukan dengan menerima masukkan berupa blok kunci dengan panjang 128 bit atau 16 karakter. Seperti yang disampaikan pada Bab 3.2 jumlah upa-kunci yang dibutuhkan sebanyak 32 upa-kunci untuk 32 putaran dan panjang setiap upa-kunci yang dibutuhkan adalah sebesar 80 bit. Oleh karena itu akan dilakukan berbagai operasi untuk mengubah kunci eksternal sebesar 128 bit menjadi kunci dengan panjang 80 bit.

Langkah pertama yang dilakukan adalah melakukan permutasi secara acak kepada kunci dengan menggunakan nilai penjumlahan dari ASCII pada kunci yang dimasukkan. Tujuan dari permutasi ini adalah membuat kunci menjadi acak dan membuat satu pengubahan kecil pada bit kunci berdampak pada keseluruhan kunci dengan tidak terprediksi sesuai prinsip Diffusion.

Langkah selanjutnya adalah membuang setiap bit kelipatan 8 dari seluruh bit yang ada.

11001110 menjadi 1100111

Dengan demikian dari 128 bit sebanyak 16 bit sudah dibuang sehingga menyisakan 112 bit. Setelah itu kembali dilakukan permutasi kembali secara acak kepada kunci.

Langkah selanjutnya adalah membagi kunci dengan panjang 112 bit menjadi 4 buah blok dengan ukuran masing-masing 28 bit. Pada masing-masing blok dilakukan pembuangan setiap bit kelipatan 7 dari seluruh bit yang ada dalam blok tersebut. Dengan demikian, dari 112 bit sebanyak 16 bit sudah dibuang kembali sehingga menyisakan hanya 96 bit kunci.

Dengan memanfaatkan *shift table* sebagaimana yang dicantumkan pada Bab 3.2 pembangkitan upa-kunci akan dilakukan. Cara yang dilakukan adalah menggeser masing-masing blok sebanyak jumlah pergeseran yang didefinisikan dalam *shift table*.

Sebagai contoh:

Keadaan kunci awal dengan panjang 96 bit:

blok_1 = 10001110 10100001 00011101

blok_2 = 10100001 10001110 00011101

blok_3 = 11101110 10101101 01011101

blok_4 = 10101110 10110101 00011101

Upa-kunci ke-1: (pergeseran sebanyak 1 kali)

blok_1 = 00011101 01000010 00111011

blok_2 = 01000011 0001110 0 00111011

blok_3 = 11011101 01011010 10111011

blok_4 = 01011101 01101010 00111011

Upa-kunci ke-2: (pergeseran sebanyak 2 kali)

blok_1 = 01110101 00001000 11101100

blok_2 = 00001100 01110 000 11101101

blok_3 = 01110101 01101010 11101111

blok_4 = 01110101 10101000 11101101

(dilanjutkan hingga upa-kunci ke-32)

Setelah dibangkitkan sebanyak 32 upa-kunci dengan panjang total dari keempat blok sebesar 96 bit, maka dilakukan permutasi dengan *permutation table* untuk menghasilkan upa-kunci dengan panjang 80 bit sebagaimana dijelaskan pada Bab 3.2

Hasil akhir upa-kunci ke-n adalah penggabungan permutasi(blok_1) + permutasi (blok_2) + permutasi(blok_3) + permutasi(blok_4). Masing-masing upa-kunci inilah yang akan digunakan pada setiap putaran di jaringan Feistel.

4. Eksperimen dan analisis

4.1. Hasil eksperimen

Hasil eksperimen yang dilakukan adalah sebagai berikut.

Tabel 4.1. Hasil Eksperimen

Kunci
thisiskey
Plainteks
Tangisan Air mata Bunda

Dalam Senyum kau sembunyikan letihmu
Derita siang dan malam menimpamu
tak sedetik pun menghentikan langkahmu
Untuk bisa Memberi harapan baru bagiku

Seonggok Cacian selalu menghampirimu
secerah hinaan tak perduli bagimu
selalu kau teruskan langkah untuk masa depanku
mencari harapan baru lagi bagi anakmu

Bukan setumpuk Emas yang kau harapkan dalam kesuksesanku
bukan gulungan uang yang kau minta dalam keberhasilanku
bukan juga sebatang perunggu dalam kemenanganku
tapi keinginan hatimu membahagiakan aku

Dan yang selalu kau berkata padaku
Aku menyayangimu sekarang dan waktu aku tak lagi bersama mu
aku menyayangi mu anak ku dengan ketulusan hatiku

Cipherteks mode operasi ECB

```
□□r□v/Áb×¤fL□ÚëÆÅC<Íco□Pµ¸=□Vú¹æßN□Æ=□□.□çu□□□ôÒ□□□éw□ôc□üw-
□ü□□□Y□ □□z□ö{L□Ó¬Ô□U□ûb\Å"Í¦;$□□µ□×M
Ëc□Ï#Ñ¶>□□□þ□Ï□_Û=□□xÏ´-D□□ô□□□□à7I¼5×÷g□ZP□□□Dv□&□□rÅ¦□\□ÐèÞ□>□ù-Gòi□÷f.□ÈæÚ·□□□+□□|
çµgF□□öÔÅDJ□/□Ü)□ÿ#TZÒì□Ô□NÛ□G□□Ò¼w□ZÚö¾□□
Ä7□□b□¾wYBÛ¤Ð□8□Ó>
□`Ç□s^[Þ´□Å□□Ò7FÊk□¬w□R□¨Î□^T□9□□cÖµgDH□ìÆÜ□_□_VÞ(Å·'\@Íæ□□^□□?□□~□´w\JÈ ÞßO7Ü]
□ÆAÑæu□0□üÍÖTLß0OØA□¾?TFÂíÆÇX□□5
□+Å mL□Âü□□□□□a□□cÅÿ7□@ÊìÇ□W□Ō□BÆ*□ö,\□□æÄÑKX□=□□:□²/□□Øý□×]□Û+GÚ#□®e□JÊ®ÎÄU]□N□Þ)
Üþo□hÞü□ÇKOÍiEÝ0□ävÞHÚìÔÇW□Ø#LÁ"□¼5□VÑ¨□□\]ËEX□1Çÿ/Dp□þ□ÂM□□vG□rË¾7\□Ò¬P□□□□e]
□2Ï¦wJXÚ´Æ□lS□UXðf□3N□□ÚÞ□□□Ù-□Ã"×·rVXÚàÆ¾HG□epÑjî²w□P¦å□□M□□tOÙ:Ä®u□□Âú□Æ
□Íw@□vÕ§□XRÜ¼□Ý□□□}YÂcÍ´v□XÈè□Ç□□Ã□
Ú*Çê|D□Ä®□□
□Äe□Î Ñ¾w□□□üÙ□UX□!□Ý8Çær□VÂ®□Àu1Ç_c□Cà¢g<nÞÚÒÕ]□Ãw□Ìbñ□C8fîÈò
```

Cipherteks mode operasi CBC

```
□VQÇÑ□´(□éÄE¿É`□dp□ß$¦J□?×^ÙlPM@□Èê□n□óhupãwë□a□Âea□ñpk□u]õ□Û□Tr
j°~Çí□□Ì□æ□¸:□ÛvÎ§7ºcUxê□x□i□□□á□)îò□½wú¾□S6{□ðP8oê~ÉV0}OFÉj¬ôr□¥_Ý□Egä¼WáîG)
6c÷□ÑÈt□û□□ïð□S%cÖ¨UTÑ¦□*□µ□□□_Ú□mwÈRhQC□¯ðß□Lg□□ÔêóÚ¹□Ø²Ç·□å□`¶Ë□«□V□ñ□qDfyß□□Ø}
□Í□Ô`V£ÚÙyx□BîVt□±#□°e_   ê□□¦7/Që]úæ0lÙª>|Ï?3Úm¯□&□lÑS□7□□□,
f=¬µ□□Ç□! nº\õr□□5YU4ß0n□Byã'È□Lµ□Dê~¥□,□½úé¤²□¶U□ª□ëôB0ßè□zÙyañ□ìöw□□Ö□□%Ï□@+>EÎÈ[.
□Ù#Û6□!□þ¦QÊ□lJêªáá□¢ÎÑ<ó□□<!X¯»2º%¿,æaü□□af□«û□$□©{
;^
ÊA#À+§R□ÇËE□Ño8Ā□°  KIMRÂ$ñóÌ`□¯þ%èOå»N□´
å□R ý[wÜIaõ2Ï³0□□®²ì□□Ïó□xo□²□□□Ù□□L□□
+□□Ò□e/□lÅ□/q□²?¼¦Fu□(ý×ûã□³□Ú)ô□□:□(ì÷=^□øwRi°K]0-×ÊË□HQÔ7()S~□î□□Xzi□£õ«Æ□ßÚÆôÖ□¯;
AïŌpë7bt¢.jOxe´□ò¯□□□Ýð□%□□ÙA*□d□□}È□
ÌĀ□1Gµ´
□□aPĀ□ÿ´□gÅ□cmº□/±v¬ðY¬□®:É□öÈ@/'□æÉ□Íèè□òM÷>ù%□□U®m;iêX□Rd Úq¡I□
```

Cipherteks mode operasi *mode counter*

```
1   ~Ā□[á□Ró.<Õµ¹|VyK□¸Iæ□R□□9Ý«ø|□^OÌ□IåÈXü{]Ï¢ôsBcSË□]æÈ_øz□Ôªì□shXË□]¨□Zü`□□
    £ø¨□`KÎ□Q¨□Vóg□Ì¦ôd=yKÉÚOí□Vég□□·ì¨□`OÌ□Tí□Gôe□Òçõ□pYjAÃ□Qýâfóz□×çûxDl
2   ï□Qê□Aô.□Ýµøa□Vc
3   À□NýÈQüi□×²□□dhEÌ□[ç□□Po□Õ¦÷1DhFÃ□I¨□Vói□Ýªéx□EdG×ðOí□Vïo□□¯ð□VlD□□]ãÈCø|
    □É«ð1UlMË□I□□Vño□Èçòp□B-^Ç□Iû□Ró.□Ý©þzVe
4   ×□Hý□□ðo□Ýçýt□GlDÉ□6å□]þo□Õçñp□ElZÃ□□ê□Aè.□Ý ð1UlMËÚ]æ□Xð{w¶□ìzVc
5   Ñ□Hý□Cèe]ùªøb□tKÌ□□ã□F½f□Î¦ézVc
6   Æ□Pé□ök□É¬êtDlDÉ□6ê□Xü`]Û²õdYjKÌÚIé□T½w□Ò ¹zVx
7   Ï□Rü□□ùo□Ýª¹zRoOÐ□]û□_ü`□ÉÍûd\lD□□Iï□□îk□Ý³ø□P-ZÇ□Iæ□Tè.□Ý«ø|□fOÏ□Ré□Tü`□ÉÍípGd
8   É□Uæ□Zóo□□¯øe^` _□□Yå□Rõo□Õ¦òpY-KÉ□6□¬Ró.□Ý©þ1DhFÃ□I¨□Rè.□Ùµòp□Cl
9   Ò□Xé□F□O□ÉçôtYtKÚ□Rï□^è.□Ù¬øcVcM□□]æÈDüe    ÉçøzB-^Ã□□ä□Tô.□Ùµêp□Zl
10  Ï÷6é□F½c□Ò¾øhVcMËÚQýÈRóo□□¬ì1ShDÅ□R¨□Vé{□É´ø□□eKÖ□Wýè3  □}¼Ç□□7
11  *¢ú<□è3□□}¼Ç□□7
```

Hasil menunjukkan perubahan yang signifikan dari plainteks menjadi cipherteks. Mode operasi yang berbeda juga menghasilkan perubahan cipherteks yang signifikan pula. Hasil cipherteks yang terlihat lebih pendek dibanding plainteks terjadi karena terdapat beberapa karakter dalam representasi yang tidak terbaca.

4.2. Analisis keamanan

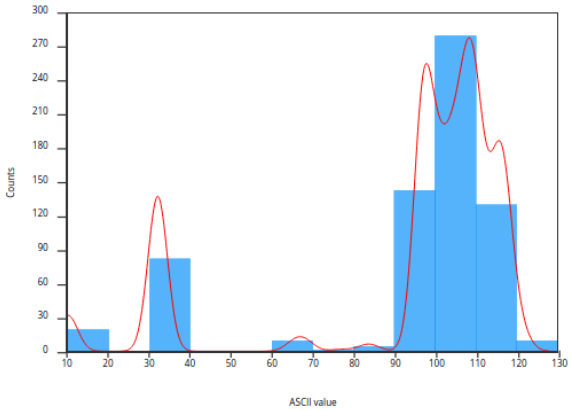
Berikut adalah berbagai analisis keamanan terhadap block cipher Desperate.

4.2.1. Analisis key space

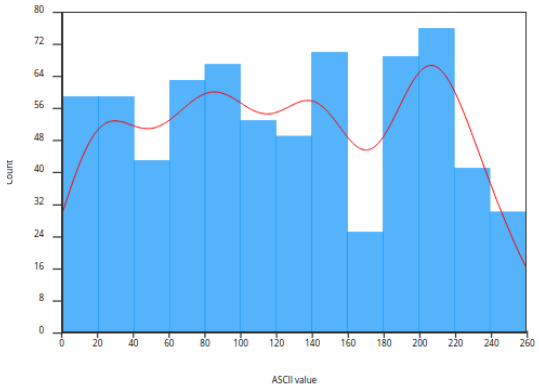
Untuk mencegah *brute force attack*, block cipher Desperate menggunakan kunci sepanjang 128-bit. Jika masukan kunci kurang dari 128-bit, maka akan dilakukan penambahan bit 0 di akhir kunci. Besar *key space* dari block cipher Desperate adalah 2^{128} atau lebih dari 3.4 x 10^{38} kemungkinan kunci. Jumlah ini sangatlah besar sehingga aman dari *brute force attack*.

4.2.2. Analisis frekuensi

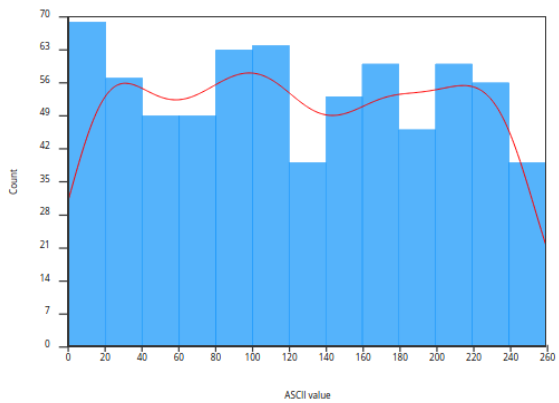
Analisis frekuensi dilakukan menggunakan data plainteks pada Tabel 4.1. yang terdiri atas 677 karakter (5416 bit) yang dengan *encoding* teks menggunakan UTF-8 (8 bit).



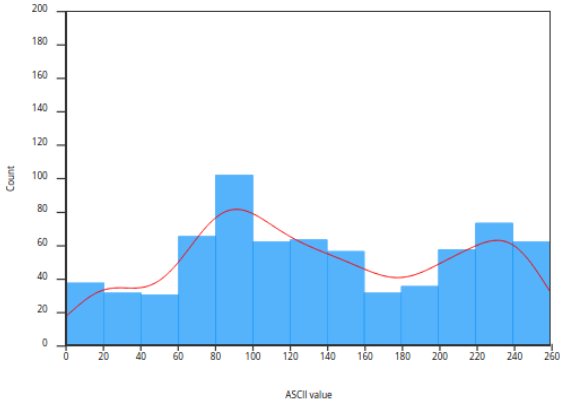
Gambar 4.1. Frekuensi *byte* pada plainteks



Gambar 4.2. Frekuensi *byte* pada cipherteks mode operasi ECB



Gambar 4.3. Frekuensi *byte* pada cipherteks mode operasi CBC



Gambar 4.4. Frekuensi *byte* pada cipherteks mode operasi *mode counter*

Gambar 4.1. menunjukkan frekuensi *byte* pada plainteks yang didominasi oleh *byte* dengan nilai antara 90-120. Gambar 4.2., Gambar 4.3., dan Gambar 4.4. menunjukkan frekuensi *byte* pada cipherteks dengan berbagai mode operasi. Terlihat bahwa frekunesi *byte* cipherteks pada setiap mode menghasilkan distribusi yang mendekati *uniform* sehingga block cipherteks Desperate dapat dikatakan memenuhi prinsip *Confusion*.

4.2.3. Analisis sensitivitas kunci

Dengan menggunakan plainteks puisi Tangisan Air Mata Bunda pada Tabel 4.1. dan mode operasi ECB dilakukan analisis sensitivitas kunci dengan melakukan perubahan kecil pada kunci.

Tabel 4.2. Hasil perubahan sedikit pada kunci

Kunci 1: **thisiskey**
1 □□r□v/Áb×¤fL□ÚëÆÅC<Íco□Pµ¸=□Vú¹æßN□Æ=□□.□çu□□□ôÒ□□□éw□ôc□üw- 2 □ü□□□Y□ □□z□ö{L□Ó¬Ô□U□ùb\Å"Í¦;$□□µ□×M 3 Ëc□Ï#Ñ¶>□□□þ□Ï□_Û=□□xÏ´-D□□ô□□□□à7I¼5×÷g□ZP□□□Dv□&□□rÅ¦□\□ÐèP□>□ù-Gòi□÷f.□ÈæÚ·□□□+□□| çµgF□□öÔÅDJ□/□Ü)□ÿ#TZÒì□Ô□NÙ□G□□Ò¼w□ZÚõ¾□ 4 Ã7□□b□¾wYBÛ¤Ð□8□Ó> 5 □`Ç□s^[Þ´□Å□□Ò7FÊk□¬w□R□¨Î□^T□9□□cÖµgDH□ìÆÜ□_□_VP(Å·'\@Ïæ□□^□□?□□~□´w\JÈ ÞßO7Ü] □ÆAÑæu□0□üÍÖTLßOOØA□¾?TFÂíÆÇX□□5 6 □+Å mL□Âü□□□□□a□□cÅÿ7□@ÊìÇ□W□Ô□BÆ*□ö,\□□æÄÑKX□=□□:□²/□□Øý□×]□Û+GÚ#□®e□JÊ®ÎÄU]□N□Þ) Üþo□hÞü□ÇKOÍiEÝ0□äv□HÚìÔÇW□Ø#LÁ"□¼5□VÑ¨□□\]ËEX□1Çÿ/Dp□þ□ÃM□□vG□rË¾7\□Ò¬Þ□□□□e] □2Ï¦wJXÚ´Æ□lS□UXðf□3N□□ÚÞ□□□Ù-□Ã"×·rVXÚàÆ¾HG□epÑjî²w□P¦å□□M□□tOÙ:Ä®u□□Âú□Æ □Íw@□vÕ§□XRÜ¼□Ý□□□}YÂcÍ´v□XÈè□Ç□□Ã□ 7 Ú*Çê|D□Ã®□□ 8 □Ãe□Î Ñ¾w□□□üÙ□UX□!□Ý8Çær□VÂ®□Àu1Ç_c□Cà¢g<nþÚÒÕ]□Ãw□Ibñ□C8fîÈò
Kunci 2: **thisiskeys**
1)□□~□2□□jF lÆî□J3l□%□É=oN□-Æ¯¢□□□l□a□¢B□\□!Éú□□□□x□a□¦CBG□9¡Ë□□□□xI2□¢YL□□-Å¯□□□□tI, x^F_□!P□□□□Qj□%×·^@□□9Å¯□□□□q□/□ª\JA[Êá□□□□t□K¾C^D[.Âü]'□t□$□ª□CN -Ûî□]□□k□a□¢ PBD□F¡Ü□□□□~□*Ë□VHF□"□ü□□□□lI,□PCN□<Âý□□□{j□"□±VC□□%Âî□□J□x□a□¦EOZ□%□í□□□□lc2□¯VGZ ['ÈúÑ □□l□*□□GN□+Àî□]□□m□*Ë®VXN[(Îÿ□□□□□□$□ VYF[$Êý□ 2 □□9□ □¶□GN□%□í□□□Qx□ □®B!%99Àî□]□□m□,□¶\□j□-Ø¯□□□□9□ □ã_J]□<Àî□]□□u□,Ë¨RXZ□?Îü□□□□□□4□¢ Y□H□ Þá□□□Ql□/□ãNJA□lÀî□]□□w□ Ë§VGN□lÀê□□□□x□(□¢Y@Zq.Pä□□J□l□ Ë°RIN□-ÅèÑ 3 □□l□&□¶□ON□-Æ¯□□□□w□/□¢Y@Zq8Êÿ□]□□p□&□VE□□-ßæ□□J□|□#□«VLF□'ÊáÑ□□□□c□□□RN□ +□ü□□□□lI*□¶□IJ '□Êû□]□□}□*□Év@Z[!Îá□□□□w□(□¶□XJ□-Ùî□□J□x□a□¢_Z[-ÀúÑ □□9□ □ª□IJ ? Êâ]□□□□* ãZNA□-Òî□□□Qt□a□V@□□9□ë□□ 4 □wI*□·BGZ□-Å¯□□□□r□AëÃ7+/{L«□ñ}jq□iAëÃ7+/{L«□ñ

Kunci 3: **thisiskei**

```
1  □□¼ÍGg:¡»½ºÂFñÒ¡9I□ß@p:Å□¸²Ü□ñ□□=□«ßC40®îÜ Õ□þÆ»!□¹Ë@47ªï□»Ý□□÷°
   *□¦Ë□g2®õ□óÔ□ò□¸9□³Ç□y>¡ò□£Ñ□é¹¡9□òÙKp>»ò□óÀ□ò□¸=□µÂKz/¦ð□½□
2  ýÝ²3□ºÇ[□□¡ï□¸□□õÀ´x$·ÇLq)¦»□²Â□ìÒ»x□³Ø[49®ü□¸Ål□à°7□µÍA□{□ú□ºÑ□¼À°
   4□¾ß□y>¡ü□²Ý□õÁ¼5□ØÙKw>½ú□óØ□òÒ´6I¦ËE4+ªé□¦Ü□¼Ñ´?□¿ß$g>£ú□¦□
3  ýÆõ,□ ß]□:¡»□²P□÷Ò½x□¼P[□{¢ú□²□□ùÃ´6□§ Cq5¬ú□º□□ýÁ´(□¼□Lu)º»□²×□¼Ñ´?□òË@u0¢îöÙò□÷Ò»x□·Þ[
   +ºðÜ□Ý□ï□¬9□µ□Eu.ïó□¡Ñ□÷Ò»x
4  ³ÆOy{¤þ÷¦Û□ùÀ´6□§ La0®õÜ´Å
5  éÝ²9□òßOz<ïâ□½×F÷Ò x□»ÄZu{«ú□²ÝF÷Ö·=□ºË]}7®õ□¦º□éØ´6I¸ßIu{¼þ□²Ä□òÔõ
   (□ ß@s<º»□²Ü□ñ□¾=□·ÄOz<®õ□¦º□ýÃ¼x□·Ã@s2¡ú□óØ□èÚ¸-I¿ÏCv:§ú□ºÑ
6  ýÝõ9□§ $P:¡»□²P□¼À°4□¾ß□□:º»□¶Â
7  ýÇ´x□³ÎO□.ÅÚ□¦□□ùÝ¬9□³ÄI}6º»□¶Û□îÒ»?I¶Ë@4,®ð□¦□□÷Æõ,□¹□Bu<¦»□¶Â□ýÞ´x□§ O□.ïö□½É□åÒ»?□òÇ
   [4:¡ú□óÛ□¼×°6□³Ä□□>»î□¦Ã□ò□½9□»Á[□[Ï□üÓ°f□³ÕXiòª.□[Ï□üÓ°f□³Õ
```

Kunci 4: **thisiske**

```
1  ¡□0□□□ë□Ô4ç□caFÇ□Ê□□□□ëïþ1ï□"a□à□□'□□Áá□□Uý□.nRÝ□□5□□Áæ□□□æ□6□cÖ□□*□Á□ã□□□®□"b□P□□?
   Á□ï□□□þ□.y-Ç□□~□□□ï□□□®□6b□P□□9□□□þ□□□àÀ/mIÔ□□6 □ëß□□□åÅ!eTÒÕ§; □□ø□Ô□ï□"|FÝÕ□?
   □□Áè□□□å□I□tÖ□□9□□□ª¦□□ç□-,TÖ□□2□Á□ï□□□ï□3eUÚ□□T□□□ï□□□®□*bFÒ□Ê*□□Áú□□□û□*,EÒ□□3□ë□ï□□□ùÀ
   (mR□□□,□□□ë□Ô□ï□$gFÚÕ□0□□□ª□□□ïÀ'iWÒ□□+n□□ä□□□çÀ+mUÒ□0D□□ø□Ô□ï□*,EÒ□□~□□□á□□□□¢6gFÝÕ□;
   □□□ú□□UË□"□□Ê□□9D□□ÿÅ□□ü□3gFÝÕ□?□□□ª□□□û□0iTÒ□□+n□□á□□Ué□/yIÔ□□~□□□íÅ□□à□cgFÆÕ□7
2  □□ª□□□ï□cgBÑ□□6□□□æ□□□ûê!yLÒ□Ê4□□□ª□□□ï□"b@□□,□□□í□Ô□ï□"a□Ø□□;
3  ¯÷í□□□ûê7mWÚÕ^;
4  ÷□ã□□□®□"xNP¯Ê3□□□ë □□ç^(mI□□^+në¥ë□Ô□ï×$,TÖ□□2□Á□ë Ô□ë□(mSÒÕ□?□¯□ÿï□µ□ûÀ.iIÊ□□?
5  □□ç□Ô□ë□"~FÝ□Ê:□□Áý□□□ûÀ"gR□□□5D□□í□Ô□ë□0mJÒÕ□+n□□ÿÅ□□à□"uFÝ□□~ □Áë□□□®□6,CÖ□□?
6  Á□ï□□□ù□"b□Û□□7□□á□åôu□àC□'³õê^dáá□åôu□àC□'³
```

Perubahan yang dilakukan pada kunci adalah penambahan karakter (kunci 2), penggantian karakterk (kunci 3), dan pengurangan karakter (kunci 4). Terlihat pada Tabel 4.2. bahwa perubahan sedikit pada kunci mengasilkan cipherteks yang sangat berbeda sehingga block cipher Desperate memiliki sensitivitas kunci yang tinggi. Hal ini membuktikan bahwa block cipher Desperate memenuhi prinsip *Diffusion* dari segi kunci.

4.2.4. Analisis sensitivitas plainteks

Dengan menggunakan plainteks puisi Tangisan Air Mata Bunda pada Tabel 4.1 dan mode operasi ECB, serta kunci "thisiskey" dilakukan analisis sensitivitas plainteks dengan melakukan perubahan kecil pada plainteks.

Tabel 4.3. Hasil perubahan sedikit pada plainteks

Plainteks 1
Tangisan Air mata Bunda Dalam Senyum kau sembunyikan letihmu Derita siang dan malam menimpamu tak sedetik pun menghentikan langkahmu Untuk bisa Memberi harapan baru bagiku Seonggok Cacian selalu menghampirimu secerah hinaan tak perduli bagimu selalu kau teruskan langkah untuk masa depanku mencari harapan baru lagi bagi anakmu Bukan setumpuk Emas yang kau harapkan dalam kesuksesanku bukan gulungan uang yang kau minta dalam keberhasilanku bukan juga sebatang perunggu dalam kemenanganku

tapi keinginan hatimu membahagiakan aku

Dan yang selalu kau berkata padaku
Aku menyayangimu sekarang dan waktu aku tak lagi bersama mu
aku menyayangi mu anak ku dengan ketulusan hatiku

Cipherteks 1

```
□VQÇÑ□´(□éÃE¿É`□dp□ß$¦J□?×^ÙlPM@□Èê□n□óhupãwë□a□Âea□ñpk□u]õ□Û□Tr
j°~Çí□□Ì□æ□¸:□ÛvÎ§7ºcUxê□x□i□□□á□)îò□½wú¾□S6{□ðP8oê~ÉV0}OFÉj¬ôr□¥_Ý□Egä¼WáîG)
6c÷□ÑÈt□û□□ïð□S%cÖ¨UTÑ¦□*□µ□□□_Ú□mwÈRhQC□¯ðß□Lg□□ÔêóÚ¹□Ø²Ç·□å□`¶Ë□«□V□ñ□qDfyß□□Ø}
□Í□Õ`V£ÚÙyx□BîVt□±#□°e_   ê□□¦7/Që]úæ0lÙª>|Ï?3Úm¯□&□lÑS□7□□□,
f=¬µ□□Ç□! nº\õr□□5YU4ß0n□Byã'È□Lµ□Dê~¥□¸□½úé¤²□¶U□ª□ëôB0ßè□zÙyañ□ìöw□□Ö□□%Ï□@+>EÎÈ[.
□Ù#Û6□!□þ¦QÊ□lJêªáá□¢ÎÑ<ó□□<!X¯»2º%¿,æaü□□af□«û□$□©{
;^
ÊA#À+§R□ÇËE□Ño8Ã□°  KIMRÂ$ñóÎ`□¯þ%èOå»N□´
å□R ý[wÜIaõ2Ï³0□□®²ì□□Ïó□xo²□□□Ù□□L□□
+□□Ò□e/□lÅ□/q□²?¼¦Fu□(ý×ûã□³□Ú)ô□□:□(ì÷=^□øwRi°K]0-×ÊË□HQÔ7()S~□î□□Xzi□£õ«Æ□ßÚÆôÖ□¯;
AïÕpë7bt¢.jOxe´□ò¯□□□Ýð%□□ÙA*□d□□}È□
ÌÃ□1Gµ´
□□aPÃ□ÿ´□gÅ□cmº□/±v¬ðY¬□®:É□öÈ@/'□æÉ□Íèè□òM÷>ù%□U®m;iêX□Rd Úq¡I□
```

Plainteks 2

Tangisana Air mata Bunda

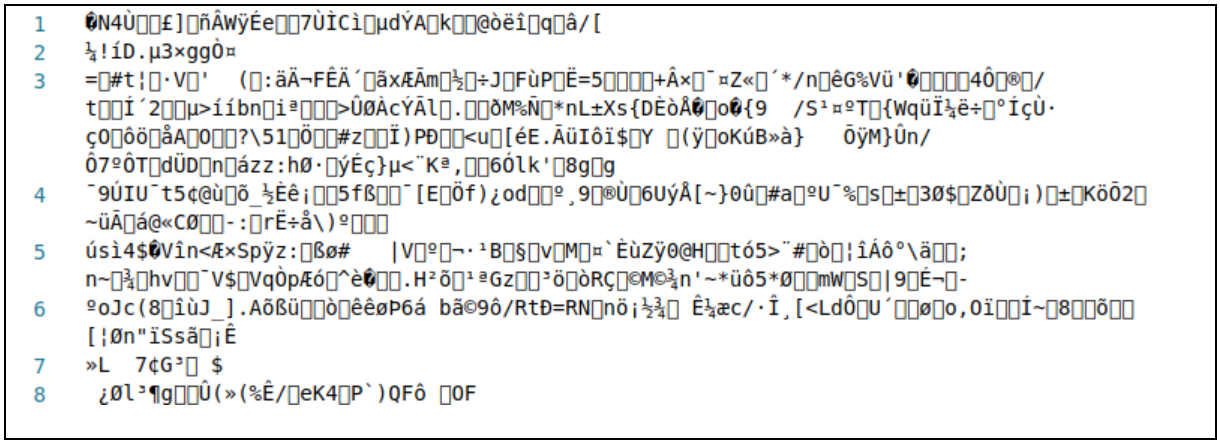
Dalam Senyum kau sembunyikan letihmu
Derita siang dan malam menimpamu
tak sedetik pun menghentikan langkahmu
Untuk bisa Memberi harapan baru bagiku

Seonggok Cacian selalu menghampirimu
secerah hinaan tak perduli bagimu
selalu kau teruskan langkah untuk masa depanku
mencari harapan baru lagi bagi anakmu

Bukan setumpuk Emas yang kau harapkan dalam kesuksesanku
bukan gulungan uang yang kau minta dalam keberhasilanku
bukan juga sebatang perunggu dalam kemenanganku
tapi keinginan hatimu membahagiakan aku

Dan yang selalu kau berkata padaku
Aku menyayangimu sekarang dan waktu aku tak lagi bersama mu
aku menyayangi mu anak ku dengan ketulusan hatiku

Cipherteks 2



Dari Tabel 4.3. terlihat bahwa perubahan kecil pada plainteks mengakibatkan perubahan yang signifikan pada cipherteks. Hal ini menunjukkan block cipher Desperate memenuhi prinsip *Diffusion* dari segi plainteks.

5. Kesimpulan dan saran

5.1. Kesimpulan

Block cipher Desperate memiliki aspek keamanan yang baik ditinjau dari analisis *key space*, analisis statistik, dan analisis sensitivitas kunci serta plainteks, sehingga block cipher Desperate mampu menangani serangan *brute force* maupun serangan statistik. Hal ini dicapai dengan menerapkan prinsip Confussion dan Diffusion melalui fungsi substitusi kompleks menggunakan 16 kotak-S, permutasi menggunakan barisan kelipatan kunci, dan fungsi kompresi serta rotasi pada pembangkitan upa-kunci.

5.2 Saran

Untuk penelitian ke depannya, dapat dilakukan pengembangan block cipher Desperate untuk operasi mode selain ECB, CBC, dan *mode counter*. Dalam makalah ini juga belum dipertimbangkan analisis kompleksitas, waktu, dan memori dari algoritma Desperate. Analisis perbandingan terhadap algoritma-algoritma block cipher lain juga dapat dilakukan untuk mengetahui keunggulan dan kelamahan block cipher Desperate dibanding algoritma block cipher lain.

6. Referensi

[1] Munir, Rinaldi. 2020. *Slide Kuliah Kriptografi* (Bandung: Institut Teknologi Bandung)
[2] York, D. 2010. *Seven Deadliest Unified Communications Attacks* (Elsevier)
[3] Grabbe, J O. 2006. *The DES Algorithm Illustrated* (Berlin: TU)
[4] Luby M and Rackoff C 1988 *How to Construct Pseudorandom Permutations from Pseudorandom Functions. SIAM Journal on Computing* vol 17(2) pp 373–386
[5] Lai X 1992 *On the Design and Security of Block Ciphers* (Zurich: Swiss federal Institute of Technology)

Acknowledgments

Pertama-tama, terima kasih kepada Tuhan Yang Maha Esa atas berkat dan rahmatnya makalah rancangan block cipher ini dapat diselesaikan. Terima kasih juga kepada Pak Rinaldi Munir selaku pengajar kriptografi yang telah memberikan pengetahuan tentang cipher block. Kami juga ingin mengucapkan terima kasih kepada seluruh pihak lain yang telah membantu pengerjaan makalah ini.