Algoritma *Block Cipher* NH2

Roland Hartanto
13515107
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
Jl. Ganesha 10 Bandung 40132, Indonesia
13515107@std.stei.itb.ac.id

Dicky Novanto
13515134
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
Jl. Ganesha 10 Bandung 40132, Indonesia
13515134@std.stei.itb.ac.id

Abstrak—Algoritma *block cipher* kunci simetri banyak digunakan untuk menjaga kerahasiaan informasi. Beberapa algoritma *block cipher* berhasil dipecahkan sehingga tidak aman lagi untuk digunakan. Agar sebuah algoritma enkripsi dan dekripsi aman digunakan, algoritma tersebut harus menerapkan prinsip *confusion* dan *diffusion* untuk mempersulit penyerang dan kriptanalis. Algoritma *block cipher* NH2 dirancang dengan melakukan iterasi sebanyak 16 kali dan menggunakan tiga jaringan feistel dalam setiap iterasinya. Algoritma ini juga menggunakan operasi substitusi dan permutasi untuk mengenkripsi dan mendekripsi pesan. Pembangkitan kunci internal untuk setiap jaringan feistel menggunakan operasi perkalian matriks, operasi aritmatika modulo, dan proses rotasi baik dalam bit maupun byte.

Kata kunci—kriptografi, jaringan Feistel, *block cipher*, operasi matriks, enkripsi, kunci simetri

I. PENDAHULUAN

Sejak dahulu kala, kriptografi telah digunakan untuk mengamankan kerahasiaan suatu informasi atau pesan. Algoritma kriptografi klasik yang digunakan dahulu kala memiliki dua teknik dasar utama dalam melakukan enkripsi dan dekripsi, yaitu, teknik substitusi dan transposisi atau permutasi. Algoritma kriptografi klasik terus dikembangkan. Namun, seiring dengan perkembangannya, banyak orang juga yang mencoba untuk melakukan serangan terhadap algoritma-algoritma tersebut. Algoritma kriptografi klasik pun dikembangkan menjadi algoritma kriptografi modern yang saat ini banyak dipakai seperti AES, DES, 3DES, dan lain-lain. Kriptografi saat ini juga tidak lagi hanya berfungsi untuk menjaga kerahasiaan suatu informasi. Layanan yang disediakan kriptografi saat ini adalah kerahasiaan, integritas data, otentikasi, dan nirpenyangkalan.

Agar suatu *block cipher* sulit dipecahkan, *block cipher* tersebut harus menerapkan *confusion* dan *diffusion*. Kedua prinsip tersebut digunakan untuk mempersulit kriptanalis dan para penyerang kriptografi. Namun, walaupun prinsip tersebut diterapkan, masih ada algoritma *block cipher* modern yang berhasil dipecahkan dengan serangan *brute force*, seperti DES yang dapat ditemukan kuncinya dalam waktu kurang dari satu hari.

Algoritma *block cipher* NH2 yang dirancang kali ini melakukan iterasi sebanyak 16 kali dan menggunakan tiga buah jaringan feistel pada setiap iterasinya dengan alur yang cukup rumit. Algoritma ini beroperasi secara bit maupun byte. Algoritma ini juga menerapkan prinsip *confusion* dengan melakukan substitusi dengan kotak-S yang dibangkitkan secara acak. Selain itu, algoritma ini menerapkan *diffusion* dengan melakukan operasi permutasi. Operasi lain seperti rotasi dan reverse juga digunakan dalam melakukan enkripsi maupun dekripsi. Algoritma NH2 memiliki mekanisme untuk membentuk kunci internal yang digunakan saat iterasi. Operasi yang digunakan untuk membentuk kunci internal antara lain rotasi, reverse, dan perkalian matriks.

II. DASAR TEORI

A. Prinsip Penyandian Shannon

Claude Shannon dalam karyanya yang berjudul Communication theory of secrecy systems memperkenalkan prinsip *confusion* dan *diffusion*. Dua prinsip ini sampai saat ini menjadi panduan dalam merancang algoritma kriptografi. Berikut ini adalah penjelasan mengenai kedua prinsip tersebut.

1. *Confusion*
   Confusion bertujuan membuat kriptanalis sulit mencari pola-pola statistik yang muncul pada cipherteks. Oleh sebab itu, prinsip ini bermaksud menyembunyikan korelasi antara plainteks, cipherteks, dan kunci.
2. *Diffusion*
   Prinsip ini bertujuan untuk menyebarkan pengaruh dari satu bit baik pada kunci atau plainteks pada cipherteks. Hasil yang diharapkan dari *diffusion* adalah cipherteks yang tidak dapat diprediksi hanya cukup dengan mengubah satu atau dua bit pada plainteks.

B. Block Cipher

Pada *block cipher*, enkripsi dan dekripsi dilakukan per blok bit. Plainteks dibagi menjadi blok-blok bit yang sama panjang. Hasil enkripsi pada blok bit plainteks memiliki panjang yang sama dengan blok bit plainteks. Enkripsi dilakukan menggunakan kunci berupa blok yang panjangnya sama dengan panjang blok plainteks. Block *cipher* memiliki banyak cara pengoperasian, yaitu, Electronic Code Book (ECB), *Cipher* Block Chaining

(CBC), *Cipher* Feedback (CFB), Output Feedback (OFB), dan mode counter. Berikut ini adalah penjelasan dari masing-masing cara pengoperasian *cipher* block.

1. Electronic Code Book (ECB)

   Pada ECB, setiap blok plainteks dienkripsi secara independen menjadi blok cipherteks. Keuntungan penggunaan mode ini adalah kita tidak perlu mengenkripsi file secara linear. Karena setiap blok adalah independen, kesalahan satu atau lebih bit pada blok cipherteks hanya memengaruhi blok cipherteks tersebut. Blok cipherteks lainnya tidak mendapat pengaruh dari kesalahan enkripsi pda blok tersebut. Berikut ini adalah ilustrasi dari ECB.
2. *Cipher* Block Chaining (CBC)

   Tujuan utama dari CBC adalah untuk membuat ketergantungan antar blok. CBC membuat setiap blok cipherteks bergantung tidak hanya pada blok plainteksnya, tetapi juga pada seluruh blok plainteks sebelumnya. Cara kerja CBC adalah blok plainteks yang akan dienkripsi di-XOR-kan terlebih dahulu dengan cipherteks hasil enkripsi sebelumnya. Berikut ini adalah ilustrasi dari CBC.
3. *Cipher* Feedback (CFB)

   Pada mode CFB, plainteks dienkripsi sebanyak n bit setiap kalinya. Nilai n ini bisa lebih kecil atau sama dengan ukuran blok. Cara kerja CFB mirip dengan CBC, tetapi pada CFB proses enkripsi dapat dilakukan seperti stream *cipher* dengan memperlakukan hasil enkripsi sebelumnya sebagai shift register yang Most Significant Bytenya di-XOR-kan dengan plainteks.
4. Output Feedback (OFB)

   Penerapan metode OFB mirip dengan CFB. Pada OFB, n-bit dari hasil enkripsi terhadap antrian disalin menjadi elemen posisi paling kanan di antrian.
5. Mode Counter

   Pada mode counter, counter adalah sebuah nilai berupa blok bit yang berukuran sama dengan blok plainteks. Nilai counter yang digunakan harus berbeda dari setiap blok yang dienkripsi. Pada saat awal proses enkripsi, counter diinsialisasi terlebih dahulu dengan sebuah nilai untuk mengenkripsi blok pertama. Kemudian, untuk melakukan enkripsi pada blok-blok selanjutnya, counter ditambah nilainya satu. Padding tidak diperlukan pada mode ini bila ukuran blok plainteks kurang dari ukuran blok counter. Bila panjang blok plainteks terakhir adalah x, hasil dari enkripsi terhadap counter diambil x buah bit paling signifikan untuk di-XOR-kan dengan blok plainteks yang panjangnya x.

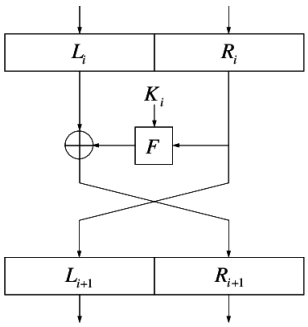
C. Jaringan Feistel

Jaringan Feistel saat ini banyak digunakan dalam algoritma kriptografi seperti DES, LOKI, FEAL, Lucifer, Blowfish, dan lain-lain. Model ini bersifat reversible sehingga dapat digunakan untuk proses enkripsi dan dekripsi. Pada jaringan Feistel terdapat fungsi f yang berperan dalam enkripsi atau dekripsi plainteks. Sifat reversible ini tidak bergantu pada fungsi f. Pada fungsi f biasanya terdapat kotak S. Kotak S adalah matriks yang berisi substitusi yang digunakan untuk memetakan satu atau lebih bit dengan satu atau lebih bit lain. Kotak S dan fungsi f dapat ditentukan sesuai dengan keinginan perancang algoritma. Oleh sebab itu, untuk membuat algoritma blok *cipher* yang kuat, perancang dapat merancang fungsi f dan kotak S serumit mungkin.

Jaringan Feistel memiliki sifat reversible. Hal ini ditunjukkan dengan persamaan berikut ini.

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

Berdasarkan persamaan di atas, fungsi XOR membuat jarigan feistel ini bersifat reversible. Agar dapat memperoleh L_{i-1} dari hasil XOR L_{i-1} dengan $f(R_{i-1}, K_i)$, cukup dengan melakukan XOR pada hasil XOR sebelumnya dengan $f(R_{i-1}, K_i)$. Hal ini juga yang menyebabkan fungsi f bisa dibuat serumit mungkin.

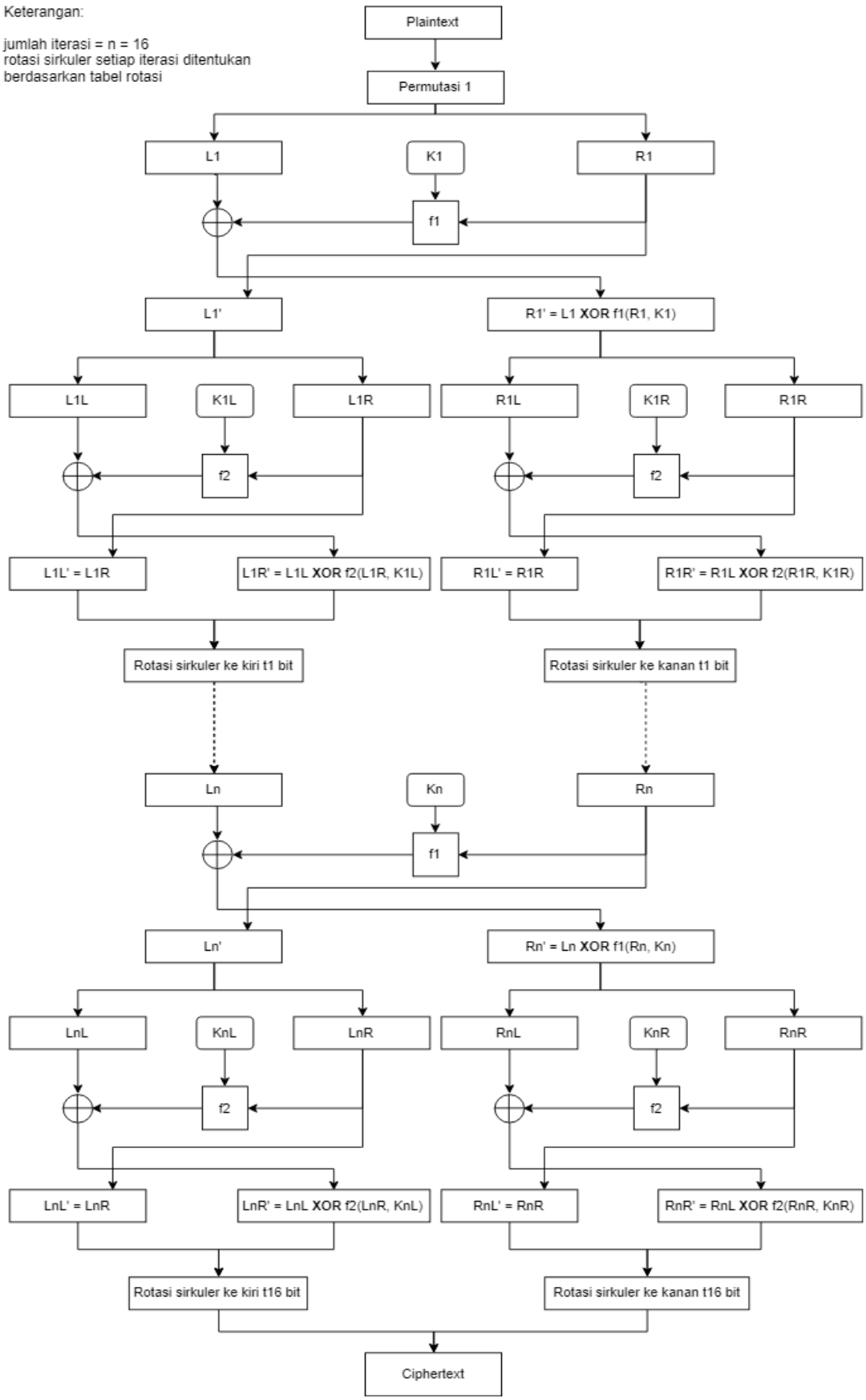


Gambar 1. Jaringan Feistel (sumber: https://www.researchgate.net/figure/Illustration-of-a-round-in-a-Feistel-network_fig1_224645711)

III. RANCANGAN ALGORITMA

A. Algoritma Secara Keseluruhan

Algoritma blok *cipher* NH2 dirancang untuk melakukan iterasi sebanyak 16 kali. Setiap iterasi, terdapat tiga buah jaringan Feistel. Jaringan Feistel pertama digunakan untuk mengenkripsi blok pesan yang berukuran 16 byte. Jaringan Feistel ke-2 dan ke-3 digunakan untuk mengenkripsi blok pesan dengan ukuran 8 byte. Berikut ini adalah alur dari algoritma enkripsi NH2.



Gambar 2. Alur algoritma enkripsi NH2

Masukan plainteks untuk algoritma ini berukuran 256 bit atau 32 byte. Ukuran kunci untuk algoritma ini adalah 16 byte untuk blok pesan yang berukuran 16 byte dan 8 byte untuk blok pesan yang berukuran 8 byte. Algoritma ini beroperasi baik dalam bit maupun byte. Operasi bit dan byte dilakukan pada jaringan Feistel dan pembangkitan kunci internal.

Pertama, permutasi dilakukan blok plainteks yang akan dienkripsi sesuai dengan tabel permutasi yang telah ada. Kemudian blok plainteks hasil permutasi tersebut masuk dalam 16 iterasi tahap *enciphering*. Tahap *enciphering* terdiri dari tiga jaringan Feistel. Jaringan feistel pertama digunakan saat pesan terbagi dalam dua kelompok pesan yang berukuran 128 bit (misal, L dan R). Setelah selesai pemrosesan pada jaringan feistel pertama, hasil pemrosesan L' dan R' akan dibagi dua lagi sehingga sekarang terdapat empat kelompok pesan yang masing-masing berukuran 64 bit. Setelah itu, kelompok pesan dari pecahan L' dan R' masing-masing diproses dalam jaringan Feistel ke-2 dan ke-3. Pada jaringan Feistel ke-1, digunakan fungsi f1 sebelum pesan pada kelompok R dilakukan XOR dengan pesan pada kelompok L, sedangkan pada jaringan Feistel ke-2 dan ke-3, fungsi yang digunakan adalah f2. Cara fungsi ini beroperasi sama, namun terdapat perbedaan ukuran pesan yang diolah. Pada f1, pesan yang diolah berukuran 128 bit. Pada f2, pesan yang diolah berukuran 64 bit. Setelah selesai diproses pada jaringan Feistel ke-2 dan ke-3, pesan pada L' dan R' digabungkan. Kemudian, pesan yang telah digabungkan tersebut dirotasi sebanyak ti bit. Pada hasil gabungan dari kelompok L', rotasi dilakukan ke arah kiri, sedangkan untuk hasil gabungan dari kelompok R', rotasi dilakukan ke arah kanan. Setelah selesai satu iterasi, hasil iterasi tersebut akan digunakan kembali untuk iterasi selanjutnya.

Tabel 1. Tabel rotasi pada saat enkripsi untuk setiap iterasi

iterasi	**t**	**iterasi**	**t**
1	1	9	5
2	89	10	13
3	1	11	8
4	55	12	8
5	2	13	13
6	34	14	5
7	3	15	21
8	21	16	3

B. Pembangkitan Kunci Internal

Dalam pembangkitan kunci internal, diperlukan kunci eksternal dari pengguna. Pembangkitan kunci internal diawali dengan tahap inisiasi. Kunci eksternal ini dibuat menjadi 32 byte terlebih dahulu dengan cara padding apabila panjang kunci eksternal kurang dari 32 byte atau mengambil 32 byte pertama dari kunci apabila panjang kunci melebihi 32 byte. Setelah itu, kunci dibagi 2 sama besar menjadi Ci dan Di. Kemudian Di dibagi lagi menjadi 2 bagian menjadi DLi dan DRi. Misalnya, apabila kunci eksternal yang dimasukkan adalah "Kriptografi", maka pembagian kunci dilakukan sebagai berikut.

Tabel 2. Kunci setelah dilakukan padding

Ci (bagian 1)				Di (bagian 2)				
K	r	i	p	t	o	g	r	DLi
a	f	i	K	r	i	p	t	
o	g	r	a	f	i	K	r	DRi
i	p	t	o	g	r	a	f	

Kemudian, potongan Ci digunakan untuk membangkitkan kunci besar yang berukuran 16 byte, sedangkan Di digunakan untuk membangkitkan kunci kecil yang berukuran 8 byte.

Proses pembangkitan kunci besar diawali dengan melakukan duplikasi Ci. Hasil duplikasi Ci di-*reverse* secara bit menjadi Ci'. Kemudian Ci dan Ci' ditampung dalam matriks buffer1 (untuk Ci) dan buffer2 (untuk Ci'). Nilai yang ditampung dalam matriks buffer tersebut adalah nilai bilangan ASCII dari karakter. Kemudian, dilakukan operasi perkalian matriks dengan formulasi sebagai berikut.

$$MHasil = (Buffer1)(Buffer2)$$

Hasil perkalian matriks tersebut, dikonversi kembali menjadi string menjadi kunci Ki dengan i adalah nomor iterasi yang bernilai 0 sampai dengan 15. Kemudian setelah itu, untuk iterasi berikutnya, nilai yang dipakai adalah Ci yang dirotasi secara byte ke kanan sebesar i dan secara bit ke kanan sebesar i*i.

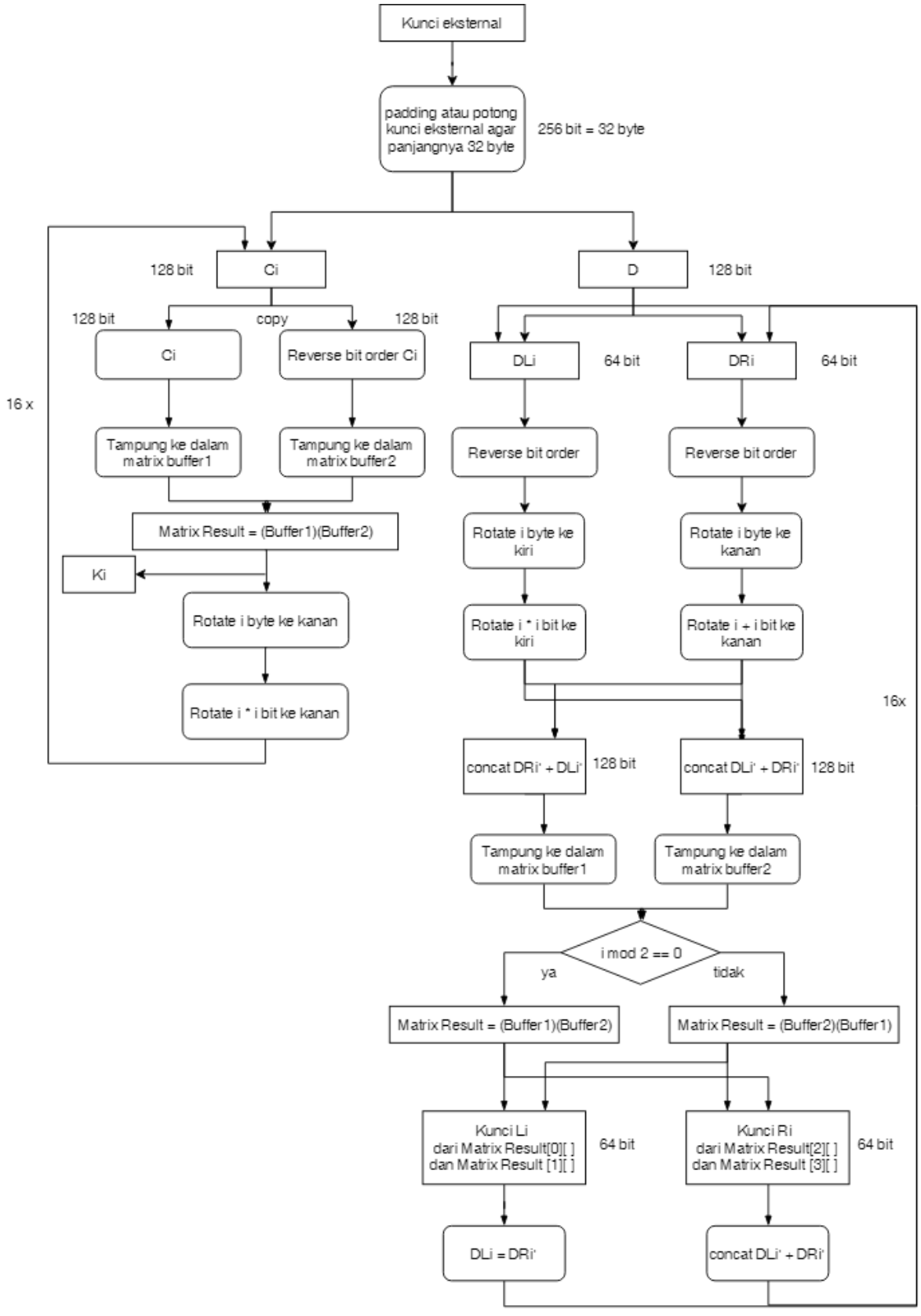
Pembangkitan kunci kecil diawali dengan melakukan reverse dan rotasi. Untuk kunci kecil pertama, dilakukan rotasi ke kiri secara byte sebesar i dan secara bit sebesar i*I menjadi DLi'. Untuk kunci kecil kedua, rotasi yang dilakukan ke kanan secara byte sebesar i dan secara bit ke kanan sebesar i+i menjadi DRi'. Definisi i sama seperti i pada saat melakukan iterasi untuk pembangkitan kunci besar. Jumlah iterasi yang dilakukan 16 kali untuk setiap kunci sehingga jumlah kunci akhir yang dihasilkan adalah 16 buah kunci besar dan 32 kunci kecil. Setelah itu, dilakukan konkatenasi kedua potongan hasil olahan kunci tersebut. Konkatenasi yang dilakukan adalah DLi' + DRi' dan DRi' + DLi'. Hasil konkatenasi tersebut dikonversi menjadi matriks buffer1 dan buffer2 dengan nilai yang ditampung adalah nilai ASCII dari karakter. Setelah itu, dilakukan operasi perkalian matriks untuk kedua buffer tersebut. Untuk i yang bernilai genap, perkalian yang dilakukan adalah sebagai berikut.

$$MHasil = (Buffer1)(Buffer2)$$

Untuk i yang bernilai ganjil, perkalian yang dilakukan adalah sebagai berikut.

$$MHasil = (Buffer2)(Buffer1)$$

Kemudian, dilakukan penukaran DLi' dengan DRi' sehingga untuk iterasi selanjutnya, DL menjadi DRi' dan DR menjadi DLi'.



Gambar 3. Alur pembentukan kunci internal

C. Round Function

Round function yang diterapkan terdiri dari 3 proses, yaitu proses substitusi untuk masing-masing byte dari pesan yang menggunakan S-Box yang berukuran 16 x 16, sehingga pengambilan 4 bit paling berpengaruh (*most significant bit*) dari 1 byte digunakan untuk menentukan nomor baris, dan 4 bit paling tidak berpengaruh (*least significant bit*) digunakan untuk menentukan nomor kolom. Isi dari S-Box digenerate secara pseudorandom dengan seed time(NULL), dan nilai output pseudorandom tersebut berada pada rentang 0 – 255.

S-Box 1

hex	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0e	31	41	7a	4f	ec	8c	a3	8e	a7	df	ce	e0	36	7c	4d
1	88	af	67	0c	19	d6	64	03	46	61	f3	a0	9d	f6	23	ab
2	27	64	25	77	50	b1	1a	df	59	f9	ad	39	2f	2a	86	b8
3	d9	ed	c4	f2	c4	28	f6	0a	89	e9	aa	26	df	cd	d1	06
4	31	f6	7d	82	a8	98	61	01	91	0e	3a	c1	38	c1	79	12
5	ae	3d	04	72	65	fa	7c	ee	e3	26	15	c2	f3	e6	c9	25
6	dd	46	a7	85	de	08	86	70	16	c0	31	4f	81	aa	61	30
7	e7	65	a2	4c	60	1f	3a	43	45	4f	06	39	36	cf	5e	13
8	15	05	98	f4	0d	1e	64	23	de	95	72	60	3f	d3	90	26
9	39	32	72	99	51	ac	dc	97	fc	e2	d0	32	b1	2e	45	c7
10	33	dd	bb	40	fb	1f	63	d9	b4	d6	39	f3	a9	c9	19	e2
11	fc	8b	7b	4d	37	58	e4	33	3a	b4	65	ec	e2	aa	b3	15
12	87	6e	55	82	8d	b9	5c	41	8f	95	34	38	5f	4d	1b	5b
13	d8	96	a8	0f	ee	8d	43	29	41	a8	15	24	53	c8	39	da
14	36	8f	5d	c3	48	b9	04	d7	4e	38	0f	ad	85	2a	08	5d
15	c1	b1	6c	af	3e	af	d8	7f	58	ed	a3	ab	b5	dd	85	eb

S-Box 2

hex	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	6c	e2	ae	b4	9b	b2	8b	ea	ea	9a	97	6f	c5	a0	cc	86
1	51	39	35	8f	e8	0e	0e	40	fb	b2	eb	b1	8f	71	9c	fb
2	53	4b	af	ef	fd	3a	d9	e8	d4	70	57	99	10	24	1f	61
3	5d	55	f0	45	63	ff	86	5e	b1	71	0f	40	e2	ac	3b	36
4	f7	ea	25	f4	24	fe	dc	f8	6e	34	92	7f	58	b1	e0	b5
5	06	d1	fa	69	d0	80	c8	81	f2	d7	c1	d4	83	fc	0a	7a
6	e6	2f	6f	0a	2d	4b	02	9c	7f	94	1b	d7	46	fb	8c	4c
7	cc	87	b6	9c	07	7e	1d	f9	55	de	ce	d9	da	d8	53	c0
8	08	c2	ca	35	0e	cd	d1	8d	61	ec	65	a7	e8	f1	f4	b4
9	78	aa	51	80	28	6e	79	7d	4d	47	56	27	20	aa	e8	28
10	6c	b2	5d	7a	7f	2f	08	e1	1b	6d	88	03	5e	7c	b8	d7
11	26	09	57	4e	77	d0	cc	c4	18	22	ec	38	cc	d4	60	39
12	86	bd	b3	06	ec	bb	e7	08	28	6f	0b	87	ec	c3	5e	12
13	cc	b5	61	44	85	2d	08	9d	4f	f4	d5	1c	c8	35	55	4f
14	f3	08	55	df	c4	3c	e7	ec	ab	f3	73	97	b6	d1	aa	83
15	86	0b	c7	0c	38	cf	a9	87	c4	7f	a3	8c	b4	f8	db	a7

S-Box 3

hex	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	01	30	87	c5	6c	6e	b1	18	61	25	af	18	f6	59	9b	7d
1	64	62	89	9c	31	32	24	f5	b1	c7	82	66	c0	5d	0d	c1
2	8e	94	86	fa	03	37	12	64	5c	c2	7c	53	1b	17	d0	80
3	79	59	1c	ab	8b	40	a0	3d	08	22	a3	c8	80	b0	89	0e
4	45	0f	08	48	46	1b	ac	a3	dd	29	f6	f8	40	c6	78	ba
5	1f	95	65	aa	d5	05	e7	dd	28	8a	a5	a8	3b	2e	b6	80
6	3d	be	c8	84	d9	74	27	b6	9d	1d	af	de	e3	27	98	02
7	bc	fd	ac	92	02	94	6f	2a	1e	15	d2	59	43	88	d9	81
8	47	a1	05	20	16	2c	d7	b3	49	86	91	2c	ad	29	2e	6a
9	26	da	fc	29	6e	6b	53	8d	80	26	e6	c4	ae	c0	45	f5
10	61	4a	16	77	76	ed	2b	bf	73	bc	eb	20	e6	19	8a	0c
11	f3	86	35	62	f2	89	ef	72	af	d5	36	5d	95	7b	53	f7
12	c5	69	6e	3b	56	99	fa	c9	56	e5	e9	3c	fe	74	48	f2
13	fa	7e	54	ec	07	43	5f	b6	18	95	13	ae	11	66	a5	d6
14	cf	13	12	25	ad	0c	ee	03	f2	d8	3f	f0	4c	87	e2	46
15	05	36	33	0c	79	92	c2	92	27	d6	40	38	3c	e5	0f	0c

S-Box 4

hex	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	f8	21	31	a5	2d	20	a8	1f	f8	e7	10	44	6f	f2	8a	74
1	29	bd	81	a2	4f	43	34	77	19	74	af	56	59	be	62	52
2	df	93	f7	0d	b3	a0	2c	ab	87	3c	ef	f6	2f	7a	6b	58
3	37	ec	fa	87	2f	2f	fe	49	a3	ad	9f	fd	6c	01	4f	4b
4	94	46	58	48	e6	85	f3	6e	c1	e3	64	f0	5d	cf	48	94
5	bb	43	1b	eb	72	19	34	15	c7	d3	12	33	d4	61	7e	68
6	a8	d7	b0	8e	5c	a4	fc	1d	87	61	0e	e4	30	56	78	ec
7	99	94	d7	0b	ad	0b	21	74	de	33	a7	b2	95	26	1a	3d
8	fd	cb	cb	59	6f	c8	76	f6	29	84	da	59	db	52	45	74
9	e6	1c	80	94	27	a1	08	05	d4	b0	b7	69	d6	d2	a6	d3
10	9d	72	2c	0c	3a	a2	02	63	27	dc	bc	02	2e	02	76	15
11	1e	f6	a9	46	97	b1	4b	6c	61	03	d5	37	d5	7c	0a	72
12	ee	36	7e	28	d9	80	8b	00	5c	47	02	8a	49	78	9f	68
13	6f	48	ae	06	fa	f9	72	5b	fc	48	93	d1	c4	9d	43	b2
14	d4	c1	da	ad	41	65	ad	9d	ac	af	28	f6	27	c7	5e	96
15	10	0c	9d	0a	05	0f	65	02	57	f8	d3	1b	96	17	cd	6a

S-Box 5

hex	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	d8	a7	17	1a	0c	c4	b7	b9	73	df	af	9a	a7	0d	31	b7
1	19	ce	c1	1e	dd	26	20	35	1f	f4	50	b5	0b	1e	1f	e3
2	c5	36	fd	d2	fa	b5	8b	6d	94	3a	07	3b	47	38	f2	60
3	06	b3	7e	e4	da	9f	19	f9	93	69	ae	9e	87	cd	81	4d
4	03	7f	1f	fd	34	aa	6a	c8	e4	71	04	2b	aa	f6	8b	b0
5	aa	09	94	84	a8	ad	7d	3b	17	2b	d9	9e	f8	5b	eb	fb
6	da	0a	f8	0e	b4	62	d6	98	d3	da	c3	7d	d1	4e	2e	7b
7	58	c2	ff	00	70	7c	3c	87	a7	15	25	9f	70	11	9a	4a
8	1b	92	58	d0	f4	2f	68	c7	09	2c	45	da	7a	73	55	d2
9	35	54	d3	a5	d0	0f	2c	77	24	52	16	95	63	b0	df	7e
10	42	38	4e	36	67	b7	fe	70	e3	43	4b	5d	b6	a0	30	eb
11	f5	03	91	c5	12	bd	3d	36	0f	53	cb	72	04	ab	f1	46
12	e3	3f	7d	4a	f6	7b	ba	d9	be	05	37	74	a6	67	5f	9b
13	6a	f0	60	7c	ae	9d	b2	bd	f1	7e	30	f5	29	21	3b	0c
14	60	b8	56	57	33	10	30	f1	16	67	65	bc	ce	c5	57	38
15	b5	b7	b4	63	55	67	21	46	e5	51	3b	0e	72	76	1a	d2

S-Box 6

hex	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	2f	70	29	62	80	5a	54	96	c1	b9	52	90	7e	a9	c8	34
1	61	7d	97	b6	e4	b8	fc	c9	09	37	d7	7b	ad	f1	4e	dc
2	61	77	3f	e1	d1	93	78	93	4c	ca	23	cb	74	eb	ff	d5
3	68	96	8b	4c	4f	87	15	58	be	ec	d4	6b	dd	22	48	3e
4	99	87	20	6b	1a	98	fe	66	62	21	31	d6	0c	30	ab	75
5	c7	36	c1	16	bd	d7	6e	7b	c3	42	e7	a1	64	2f	df	fe
6	b6	ff	69	d0	97	67	36	fa	88	68	d0	94	98	7c	09	5f
7	b2	cb	75	70	a2	e4	eb	65	26	d2	06	8b	01	e6	89	b7
8	e5	f2	87	7d	59	be	77	e1	26	47	75	be	c3	7f	1e	76
9	4a	93	e6	ec	77	d1	51	9e	a4	58	29	a5	3e	b2	5d	23
10	a4	e4	a0	fd	a2	17	de	c8	5f	53	87	22	d2	a5	98	1c
11	38	7e	08	b0	50	5a	4e	f4	b2	77	99	f0	29	f6	13	cd
12	db	b4	ca	7d	cb	a8	46	2a	fb	cd	4d	ce	72	e5	ea	aa
13	64	f3	5a	b4	4d	a8	a8	ff	1f	41	ef	48	38	02	15	13
14	b6	df	90	82	87	d6	ac	83	a3	f9	51	15	df	3b	c0	43
15	2e	1a	f7	7b	c3	9f	7a	e2	e0	69	2b	18	6c	40	2b	22

S-Box 7

hex	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	20	bc	a4	a7	92	51	2a	36	4a	7b	4b	29	b7	0b	6c	e5
1	26	63	61	e9	02	db	cb	e3	45	f6	fb	b1	37	27	d3	57
2	e3	78	fe	75	c9	29	ab	13	a4	f7	3d	5b	02	a9	41	28
3	0d	a2	11	0f	7d	dd	f2	c2	d3	ee	73	0a	15	47	61	f8
4	bf	60	6d	88	89	19	9b	2d	10	d8	89	12	82	ca	3b	8f
5	6c	4c	9e	e9	29	91	ac	fd	7f	1f	07	94	66	69	8c	25
6	c9	f9	ad	52	12	49	7f	22	21	08	35	a3	d2	70	32	3e
7	bc	d1	28	e6	62	d4	e3	e1	f3	ea	75	5a	53	01	7f	1c
8	fa	2d	6e	0d	76	ee	2f	97	f6	64	3b	c9	d4	6d	07	91
9	3e	2f	77	a0	03	5a	81	f7	44	f6	51	98	f7	d0	b4	f2
10	fd	23	ff	73	11	2e	0b	07	93	46	d0	67	b3	d8	f8	f2
11	07	6f	92	0b	c9	14	02	0e	0a	53	a6	02	23	5a	f4	21
12	7d	f3	94	8e	21	9f	96	b4	e5	66	1c	99	3e	14	8b	46
13	84	1d	51	4d	31	53	5b	3c	a6	01	3e	c9	5c	32	ea	d9
14	25	7f	68	46	1e	fe	fb	04	64	17	9d	a3	2b	28	e9	af
15	45	3a	fd	77	8d	58	b3	33	5a	f1	fc	b6	23	e7	8f	48

S-Box 8

hex	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	66	f7	8e	84	f5	89	88	5a	a0	25	fd	cc	4d	e6	7b	93
1	20	78	0a	ad	d1	bd	e0	2b	ae	dc	e1	d1	c3	70	19	29
2	68	a7	ae	5d	31	36	b7	d1	5c	b4	9d	a9	9a	19	3c	ba
3	91	46	67	62	03	47	8d	b1	24	6e	82	e7	df	9b	11	47
4	43	bf	a4	74	f5	5c	45	51	10	e3	fb	ab	fc	37	65	8d
5	7e	cd	f0	81	14	7d	33	38	ec	b5	20	cb	51	31	12	94
6	f0	b6	08	e5	12	4d	37	23	30	32	ce	2c	69	33	ba	e7
7	00	aa	69	15	27	9c	4d	13	51	6d	de	a2	9e	f0	36	8e
8	a7	3e	74	b9	8c	ab	dc	bc	dd	aa	e9	46	de	a3	2e	de
9	4d	97	f3	74	33	41	88	84	ae	66	27	4d	57	5d	db	fe
10	9c	4f	b7	28	fa	94	e4	d7	3e	cd	1e	1c	70	4c	fb	bd
11	e3	ee	32	16	2f	ba	9a	de	20	c1	2b	77	1f	06	75	bb
12	56	2d	e3	50	c1	c7	28	ff	95	46	1c	05	92	17	c3	75
13	05	f5	8b	35	af	25	13	cf	e7	3e	47	06	44	bc	c1	9a
14	e9	a4	eb	aa	6b	13	aa	00	59	c6	06	eb	dd	c9	60	e2
15	be	eb	17	6d	10	2a	3c	f7	68	83	fd	ad	40	be	47	29

S-Box 9

hex	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	62	32	d4	ce	45	7e	ce	9e	44	d4	89	21	9d	e9	03	5b
1	d4	1b	c8	e5	45	05	dc	ae	88	da	5b	c8	98	a2	f2	fb
2	d5	c6	c9	1a	44	97	b9	88	6c	42	a9	09	2c	ac	65	00
3	c7	2d	e5	0d	32	c2	bb	bb	9c	16	83	34	b8	75	2f	8d
4	3b	f8	a8	7f	90	61	07	fc	a3	b0	05	cf	5d	6a	d0	24
5	98	b5	31	ca	77	ec	85	13	02	09	48	bb	7e	77	48	ba
6	70	f0	39	00	51	41	fc	f5	f1	01	c4	4e	6c	94	73	04
7	4a	a4	ce	c1	91	54	d5	93	5d	1d	4e	db	94	97	95	04
8	87	cf	04	d9	10	00	ce	01	02	92	50	6e	27	c3	72	71
9	67	40	32	f8	94	07	8c	f1	24	da	cd	b9	71	62	bd	f9
10	31	c2	d2	41	c2	a0	43	c4	32	93	32	59	56	a4	ca	bd
11	e5	fd	b6	79	04	42	6b	29	1c	38	e2	8e	9a	9f	87	cc
12	61	59	0d	24	f9	50	e8	2b	e3	1b	85	39	bf	4f	f7	a4
13	4c	ad	1e	51	ef	89	7a	0b	c1	5c	99	5b	fb	20	27	5d
14	79	35	81	72	85	69	9e	69	84	23	a2	44	72	99	e8	bf
15	46	06	10	35	8f	8a	41	50	e6	da	ac	e1	fb	d3	3e	74

S-Box 10

hex	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	08	bf	e7	8e	29	85	f7	ad	a8	99	f1	1a	33	da	d9	79
1	e0	e9	af	70	73	f0	c0	59	ca	6c	3b	c5	40	79	3a	48
2	39	21	d6	62	a6	cd	0f	4e	67	01	68	9a	db	42	13	bb
3	2b	c2	2b	9f	b2	ec	f8	7d	58	33	42	98	ad	7c	e1	e6
4	9d	b7	48	43	85	57	91	ec	58	fa	86	33	3c	99	ef	67
5	5c	1a	06	0e	06	ff	8b	5f	32	ce	f7	df	4a	d8	c5	e8
6	90	0d	2b	15	65	bd	01	bd	b7	87	f1	f3	20	e0	5a	7c
7	fa	61	8b	01	60	16	60	92	e4	57	72	2f	30	37	17	c0
8	45	42	d5	aa	ff	d6	67	b6	5d	58	a9	7d	38	04	fa	33
9	65	85	34	c5	9b	94	57	80	eb	c9	af	1b	01	c6	db	46
10	08	b0	f0	08	86	57	be	e3	b0	68	61	e8	6c	5b	1b	d1
11	e0	4f	96	7b	e3	ed	fb	cf	b7	aa	ea	b8	70	c6	fe	79
12	76	ee	81	fd	45	3f	e0	f5	a7	41	de	13	9c	f9	e4	7c
13	49	7a	f8	2c	68	f3	fb	1f	9e	e6	d7	0e	ac	d5	87	22
14	c3	08	1f	08	48	00	fe	ef	41	dc	03	de	d5	e7	5a	1e
15	62	52	4b	ca	46	46	e9	e4	2c	c0	f2	d8	95	7a	fb	58

S-Box 11

hex	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	82	1a	60	ca	1a	5e	ba	5c	3a	bd	3a	10	a4	94	2e	06
1	e7	79	d0	2d	c0	b9	11	ec	79	03	c5	0e	7d	c0	66	00
2	da	c7	ca	f5	25	84	51	60	41	8b	70	e6	1f	9e	ec	06
3	18	bd	33	d8	76	44	c4	f0	48	89	fe	c5	49	65	c5	24
4	2c	90	19	51	14	6a	b1	56	f5	21	3c	14	c0	28	1b	d8
5	e5	4e	b0	5c	93	74	4c	db	fe	4a	a0	47	af	66	6b	db
6	f6	84	2d	0a	ee	de	60	e3	00	9c	f8	c0	c5	13	98	aa
7	61	48	06	f4	bc	52	cf	ba	9d	70	02	4c	d6	6d	28	cc
8	f2	55	d6	e0	33	37	c4	33	d3	bc	f3	98	cf	8b	43	30
9	d3	49	25	90	9c	f4	4a	39	64	4c	85	3a	ba	ad	06	ac
10	02	dd	8c	36	14	50	69	e7	0c	5d	80	db	e8	c3	0c	bc
11	0c	31	4c	a8	25	96	e1	8a	e3	67	c4	9d	14	cb	49	17
12	a8	d5	4d	bc	26	b6	a3	32	13	23	0e	fc	e6	1a	b8	f3
13	4b	04	9b	70	9a	7d	fa	7d	e4	bf	1a	f8	8a	63	0f	32
14	39	5c	ee	5f	13	91	91	26	b5	9f	22	9b	b9	da	8e	04
15	de	2a	75	79	a7	6f	f6	8b	2e	11	83	b8	74	93	ea	ad

S-Box 12

hex	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	ef	d8	0c	02	6a	9e	29	1f	3d	4b	ba	f7	26	49	fb	04
1	73	70	7d	1a	e0	74	a5	0e	85	28	c7	f9	bb	b1	a7	ab
2	8a	b3	ad	f4	51	d6	13	8f	22	cd	86	48	16	81	4c	89
3	f2	ca	a3	d2	3e	48	e0	c3	71	a7	bc	2c	59	63	d7	e3
4	17	85	d7	68	5b	ea	f7	7d	b7	7d	c5	ce	ff	12	57	f1
5	dc	fb	c3	1a	43	a3	dd	b4	4b	99	e1	a4	fd	b8	87	14
6	3d	5e	7c	99	48	74	16	ff	f1	dc	cd	f0	ee	25	e1	ca
7	20	a4	e4	63	48	c1	18	93	5a	f9	37	57	b1	be	6b	ef
8	1c	e8	88	64	5c	9e	63	4d	7a	31	3e	68	56	1f	32	76
9	c4	16	d9	0c	d7	f1	9f	32	ea	d6	89	9c	94	f5	8b	b0
10	dd	13	14	39	b1	77	86	2c	a8	c4	94	fe	e4	c7	74	a8
11	dd	4e	b4	b5	3f	53	e7	2a	29	70	c6	bd	65	51	6d	42
12	64	81	7b	15	f8	02	41	a1	c6	d6	9f	aa	9d	14	52	7a
13	62	06	2f	a1	59	16	cb	82	87	91	3f	ec	e2	ac	2f	46
14	2d	aa	5c	26	ac	9d	c7	73	73	66	1d	10	7a	70	8b	dc
15	76	ba	7e	d0	d1	49	52	58	db	92	44	bd	3e	73	04	6c

S-Box 13

hex	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	1e	60	92	ca	fd	59	3d	71	bf	5b	81	3a	cb	0c	16	41
1	c7	94	11	98	de	64	f0	b9	f6	34	76	34	a8	7a	a0	c6
2	da	32	90	d8	8b	ce	49	4b	29	ca	85	f4	d7	9b	35	9e
3	30	47	36	0e	ab	26	c7	a1	5a	3d	d5	02	b8	76	c8	92
4	a8	59	6a	34	27	b3	7f	50	7e	04	44	55	9f	79	f3	cf
5	c0	29	dd	6b	4f	a4	0c	a9	e2	e2	ac	9a	58	74	2c	00
6	cd	97	34	f4	4a	b3	44	c8	b7	88	1d	57	02	10	26	c2
7	39	04	2e	88	a8	3a	32	8a	1c	de	24	74	52	51	75	20
8	e8	a9	14	32	5d	59	fb	14	e1	18	6b	e3	29	92	a6	62
9	96	d4	eb	3e	0e	1d	c9	2b	fb	ed	9f	4d	3e	14	6d	26
10	be	82	59	1b	db	54	2f	bc	6c	9b	a0	95	2d	46	f8	c3
11	1a	e3	01	28	00	ca	53	fb	b8	f3	48	f6	07	b6	1d	c5
12	38	76	e0	13	ca	10	cf	36	ab	6f	cc	d8	b5	c4	9b	cf
13	a7	9c	f8	a7	67	4b	a2	1f	3e	ea	15	46	a0	32	0b	d8
14	a8	ec	eb	72	fc	bb	a9	a7	2a	75	7f	e0	39	1a	af	e0
15	b6	a7	87	1d	f3	29	3c	31	13	52	77	b4	84	83	8c	2d

S-Box 14

hex	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	6f	78	9f	6b	33	48	12	5d	bd	91	3d	f6	ab	ed	d6	61
1	94	5d	7f	87	86	bb	b9	9a	0d	30	4e	92	b3	da	bf	22
2	52	5e	8d	85	a7	9f	e3	64	30	20	5b	db	0d	31	3d	a2
3	8f	bc	29	15	77	e2	af	85	13	fd	17	c6	d8	d6	e9	2a
4	34	76	b0	db	16	93	40	46	b3	9b	22	c1	cc	5f	63	5b
5	1b	8c	71	92	6f	20	17	82	1e	2e	48	f6	04	31	20	39
6	a8	d0	14	be	63	54	04	17	ef	26	d8	bc	85	3b	17	a0
7	c7	88	33	36	a9	4a	b8	c7	79	01	bd	7d	32	dd	b6	da
8	ae	cb	98	11	1f	9d	28	0f	c3	00	cb	49	3b	e2	e9	03
9	6b	1c	39	14	67	f2	db	e0	f3	98	5d	25	75	14	00	23
10	df	98	35	fe	35	5d	0d	f9	5e	d8	42	99	bb	2b	9c	26
11	48	d6	3a	af	c8	15	8f	bb	ad	ec	e0	22	00	e0	46	df
12	79	7b	de	ae	d8	eb	a7	36	c4	e9	d0	7f	15	6c	a5	5d
13	42	df	0c	0a	f4	9b	c5	a1	87	a6	c3	88	86	09	67	ff
14	84	45	ae	5d	31	55	93	f5	3f	63	74	54	d0	19	b1	12
15	f8	bd	1d	ec	58	e2	8d	df	88	50	67	0f	5a	cf	0e	de

S-Box 15

hex	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	14	bc	3b	45	12	cf	3a	51	32	ae	a5	02	c7	56	15	bf
1	13	32	ab	6b	14	38	4a	9d	89	b2	ac	e3	81	ba	c1	95
2	77	fd	db	89	cc	15	da	fe	c4	7f	01	8b	d5	16	4b	e8
3	48	f6	53	5c	2f	9d	f9	b8	4f	a5	9b	d0	60	5c	66	d7
4	59	41	60	25	56	3a	24	1a	b9	25	a6	8e	3b	f1	76	83
5	e7	c9	df	16	66	d9	ce	b6	7e	69	86	de	c6	ec	b5	1f
6	2d	15	45	84	4f	69	9e	08	8e	44	96	c9	35	0c	4c	1d
7	d5	2b	33	3c	04	02	f2	83	6b	78	61	31	65	17	51	92
8	2c	96	16	7c	ff	b5	84	8d	f9	1b	56	2f	27	a2	4c	fd
9	cd	7f	39	d2	81	2b	55	ed	a3	b6	1e	08	cd	6f	9b	fa
10	05	b1	76	04	66	fa	91	60	15	e7	8f	3d	89	db	3a	57
11	5a	73	29	dc	9e	7e	c9	41	34	e7	4a	02	57	e5	fc	5c
12	96	72	61	fd	6c	f2	5d	82	da	ec	bf	63	c7	f9	ba	21
13	6c	e3	fd	0a	61	c6	4b	96	ae	95	98	05	7a	94	61	11
14	06	c2	0e	72	b5	6b	f4	8f	57	b3	f2	1e	ac	ad	3f	18
15	90	3d	22	f2	03	6e	88	b1	03	20	b6	7e	b4	18	8f	ba

S-Box 16

hex	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	da	9d	2c	8f	08	21	1e	5f	d4	11	7d	81	be	bc	99	4e
1	f9	bc	40	fd	2a	c8	ae	2d	e8	65	ab	9c	7d	3a	56	57
2	d7	83	e7	df	a4	05	3e	78	16	bb	f9	d4	78	93	23	71
3	4f	63	6e	79	2c	1d	a6	14	82	52	b1	ff	8c	07	56	64
4	8a	3d	43	2e	43	82	a7	59	3d	a0	2e	b5	33	51	27	82
5	b4	95	fb	e0	b2	a2	f5	34	f4	a6	33	80	ad	8a	e4	38
6	c7	28	66	0a	aa	0d	64	e7	ae	92	9d	e1	e3	c4	64	97
7	59	5f	78	0c	01	6d	40	f5	13	74	76	c0	fe	5a	f8	c5
8	82	5f	d0	2c	6c	34	14	1a	c6	b1	fc	a9	75	60	40	ce
9	bf	b8	da	c1	25	1b	b6	38	8f	2c	f9	8d	87	f1	52	09
10	50	22	36	bd	56	4a	d7	1c	fb	d3	c5	70	33	06	3e	f3
11	be	19	b4	e4	34	6a	1c	c3	97	15	50	1e	07	a2	27	57
12	c5	5d	14	1b	a7	ec	38	a2	bf	fd	12	f3	03	51	e6	c2
13	6a	9a	a6	9e	04	c2	61	9b	d8	b1	b9	df	53	e1	36	18
14	3e	4b	34	e6	37	6c	88	f6	69	9b	e9	6d	ec	cf	2f	56
15	69	d5	f4	6e	97	55	09	6f	06	c3	4e	59	a4	85	72	e2

Setelah melalui proses substitusi, dilakukan proses pemangkatan masing-masing nilai ASCII dari karakter pesan hasil substitusi dengan sebuah bilangan prima di antara 2 hingga 1000, kemudian dilakukan proses modulo dengan 256, agar hasil akhirnya tetap berada pada rentang nilai 0 – 255. Proses penentuan bilangan prima dilakukan dengan cara penjumlahan nilai ASCII dari kunci masukan, kemudian total penjumlahan nilai ASCII tersebut dilakukan modulo dengan jumlah bilangan prima antara 2 hingga 1000, sehingga dapat terpilih indeks bilangan prima yang akan diambil, dan bilangan prima yang diambil adalah bilangan prima ke indeks hasil perhitungan.

Pada tahap terakhir, dilakukan proses permutasi dari *block cipher* tersebut. Proses permutasi menggunakan tabel permutasi yang nilainya diantara 0 hingga jumlah karakter dalam 1 blok – 1, kemudian akan dilakukan proses pengocokan pada nilai-nilai elemen tabel tersebut.

IV. EKSPERIMEN

Pada bagian ini, akan dijelaskan hasil implementasi *block cipher* yang telah kami program. Eksperimen akan dilakukan dengan menerapkan algoritma NH2 dan menggunakan 5 teknik blok *cipher*, yaitu *Electronic Code Book* (ECB), *Cipher Block Chaining* (CBC), *Cipher Feedback*(CFB), *Output Feedback*(OFB), dan mode *counter*. Data yang disimpan adalah data berupa file eksternal yang berisi 16 S-Box dan tabel permutasi sebanyak 128 elemen dan 64 elemen. Kunci eksternal didapatkan dari input pengguna. Kunci eksternal yang akan digunakan adalah "**kriptografi itu seru**". Berikut adalah contoh input teks yang digunakan menjadi input.

Anyone who reads Old and Middle English literary texts will be familiar with the mid-brown volumes of the EETS, with the symbol of Alfred's jewel embossed on the front cover. Most of the works attributed to King Alfred or to Aelfric, along with some of those by bishop Wulfstan and much anonymous prose and verse from the pre-Conquest period, are to be found within the Society's three series; all of the surviving medieval drama, most of the Middle Engl
ish romances, much religious and secular prose and verse including the English works of John Gower, Thomas Hoccleve and most of Caxton's prints all find their place in the publications. Without EETS editions, study of medieval English texts would hardly be possible.

A. *Pengujian dengan Mode ECB*

Berikut adalah hasil enkripsi *block cipher* dengan menggunakan ECB

```
BC E9 F4 EA F1 E0 9B F2 4B EA 9B E5 DC DC DF EE
9B FD E7 7B 51 DC E9 DF 9B C8 E4 DF 20 E7 E0 9B
C0 E9 E2 E7 8F EE E3 9B 5F E4 EF F8 BF DC ED F4
9B F4 E0 F3 10 EE 9B F2 E4 E7 E7 9B 29 E0 9B E1
...
```

Hasil dekripsi pada sama seperti plainteks. Perubahan pada 1byte pada plainteks akan menyebabkan hanya adanya perubahan pada 1 blok *cipher*. Berikut adalah perubahan hasil enkripsi ketika kata pertama "Anyone" berubah menjadi "Bnyone".

```
C3 AE 9F 8C DF 7F E4 3A B8 9A AA C6 D3 9A A1 8F
9B FD E7 7B 51 DC E9 DF 9B C8 E4 DF 20 E7 E0 9B
C0 E9 E2 E7 8F EE E3 9B 5F E4 EF F8 BF DC ED F4
9B F4 E0 F3 10 EE 9B F2 E4 E7 E7 9B 29 E0 9B E1
...
```

Dari hasil enkripsi diatas, terlihat bahwa hanya 16 byte pertama yang berubah, yaitu 1 blok cipher pertama karena 1 blok cipher awal berubah, dan dapat dilihat bahwa perubahannya cukup drastis. Hal ini membuktikan bahwa prinsip diffusion telah tercapai dengan baik pada 1 blok tersebut.

Berikut adalah data waktu data waktu dari proses enkripsi dan dekripsi dari algoritma ini dengan menggunakan mode ECB.

Tabel 3. Waktu enkripsi dan dekripsi dengan mode ECB

Proses	Waktu (microseconds)
Enkripsi	333859
Dekripsi	338051

B. *Pengujian dengan Mode CBC*

Berikut adalah hasil enkripsi *block cipher* dengan menggunakan CBC.

```
5A 0B 02 10 4C 59 44 5A 4F 50 0B 42 1B 4F 3E 0B
40 57 03 38 4C 59 4F 0B 78 54 4F 3F 57 50 0B 24
32 1E 31 72 1E 5E 5D 6C 12 2E 7A 6E 7D 28 38 42
26 31 2D 36 30 3A 74 33 5D 0F 33 6A 11 2D 6B ...
```

Perubahan 1 byte pada input akan menyebabkan perubahan 1 byte pula pada *block cipher* selanjutnya. Berikut adalah potongan hasil enkripsi yang mengubah 1 karakter awal dari kata "Anyone" menjadi "Bnyone".

```
C6 47 80 5F B2 61 48 66 E6 8F AB E0 81 C7 EC DE
5F 39 89 68 62 9A 25 E8 A8 B5 E2 35 57 FF 5B 1D
46 DB 7C F8 3C C4 5E 23 53 0A 03 D5 D1 EF B3 30
28 ...
```

Pada hasil enkripsi hasil perubahan 1 karakter awal tersebut, terlihat bahwa terjadi perubahan total pada hasil enkripsi, prinsip difusi tercapai.

Berikut adalah data waktu data waktu dari proses enkripsi dan dekripsi dari algoritma ini dengan menggunakan mode CBC.

Tabel 4. Waktu enkripsi dan dekripsi dengan mode CBC

Proses	Waktu (microseconds)
Enkripsi	343620
Dekripsi	357980

C. *Pengujian dengan Mode CFB*

Berikut adalah hasil enkripsi *block cipher* dengan menggunakan CFB.

```
F4 48 03 2D 69 D5 F0 1E D6 D5 BC C8 8C 9D 23 DA
71 6E 49 C8 40 CA 4A 1C D5 44 50 08 EF 8D 49 E3
D5 4C 10 3F 21 00 5D F8 D5 19 C0 61 B7 E3 3B 28
51 84
...
```

Perubahan pada 1 byte karakter pada plainteks akan menyebabkan adanya perubahan pada 1 byte pesan tersebut, dan menjalar pada blok *cipher* yang lain

secara tidak terprediksi. Berikut adalah hasil enkripsi dari proses perubahan 1 byte pesan, yaitu dengan mengubah kata pertama "Anyone" menjadi "Bnyone".

```
8A 34 E7 9A 51 23 4C AD 37 5B 65 56 41 B4 82 6F
1B B7 2C F4 8B 63 9E D0 6C 5D B7 AA E5 5D 29 70
91 10 0A E3 33 56 90 6A B1 F6 C1 F2 AA 43 ...
```

Dapat dilihat bahwa perubahan 1 byte pesan plainteks, maka akan berefek adanya perubahan yang drastis pada hasil enkripsi tersebut, sehingga algoritma ini memiliki sifat difusi yang baik.

Berikut adalah data waktu data waktu dari proses enkripsi dan dekripsi dari algoritma ini dengan menggunakan mode CFB.

Tabel 5. Waktu enkripsi dan dekripsi dengan mode CFB

Proses	Waktu (microseconds)
Enkripsi	341566
Dekripsi	344344

D. Pengujian dengan Mode OFB

Berikut adalah hasil enkripsi *block cipher* dengan menggunakan OFB.

```
6C 7D 10 7E 7D 04 47 16 7F 7E 47 19 04 08 03 1A
47 1E 7B 03 47 08 7D 03 47 1C 00 03 03 7B 04 47
6F 5A 38 31 2E 24 2D 75 31 2E 29 3A 23 36 23 1E
75 29 3A 1D 29 24 75 28 2E 31 31 75 33 3A 75 37
...
```

Perubahan pada 1 byte karakter pada plainteks akan menyebabkan adanya perubahan pada 1 byte pesan tersebut, dan menjalar pada byte pada blok *cipher* yang lain namun posisi perubahannya tidak berpola. Berikut adalah potongan hasil enkripsi pesan dengan mengubah kata pertama "Anyone" menjadi "Bnyone".

```
AD 1E F3 DC C4 5C 41 F4 56 9D 15 56 D7 77 26 9F
65 7B 1F 29 31 BC 37 D1 52 0B 02 08 2C EF 88 D9
0D 7B B6 D1 D8 F7 C6 2E 94 DB 84 6B 52 AA 0B ...
```

Dari hasil tersebut, terlihat bahwa dengan perubahan 1 karakter, maka akan menyebabkan perubahan encrypted text yang drastis, sehingga dapat disimpulkan bahwa dengan metode pengujian Mode OFB, algoritma ini memiliki sifat difusi yang baik.

Berikut adalah data waktu data waktu dari proses enkripsi dan dekripsi dari algoritma ini dengan menggunakan mode OFB.

Tabel 6. Waktu enkripsi dan dekripsi dengan mode OFB

Proses	Waktu (microseconds)
Enkripsi	341972
Dekripsi	337700

E. Pengujian dengan Mode Counter

Berikut adalah hasil enkripsi dengan menggunakan mode counter.

```
08 04 47 16 70 59 39 30 2B 25 2C 74 7D 03 47 1C
00 6C 7D 10 7E 7D 2B 30 04 08 03 30 74 36 37 74
03 03 7B 04 47 7F 7E 47 19 1A 47 1E 7B 03 47 30
2B 28 37 26 33 26 1B 74 28 37 1C 28 25 74 29 3A
...
```

Perubahan pada 1 byte pesan akan mengubah hanya 1 blok *cipher* yang mengalami perubahan byte. Berikut adalah potongan hasil enkripsi dengan mengubah 1 karakter awal dari kata awal, yaitu "Anyone" menjadi "Bnyone".

```
B6 0D 4B 0E 68 B8 2A 19 85 48 46 03 02 36 FF 52
07 48 48 94 61 3B 44 1F 5E 81 D1 C9 5F 6E C8 16
7B 13 24 E3 CC 4E FC 51 97 42 55 99 79 54 EB ...
```

Perubahan pada 1 byte ini ternyata mengubah banyak nilai hexa hasil enkripsi, sehingga pada kasus ini, terbukti bahwa algoritma ini mempunyai sifat difusi yang baik.

Berikut adalah data waktu data waktu dari proses enkripsi dan dekripsi dari algoritma ini dengan menggunakan mode Counter.

Tabel 7. Waktu enkripsi dan dekripsi dengan mode counter

Proses	Waktu (microseconds)
Enkripsi	343102
Dekripsi	336588

V. ANALISIS KEAMANAN

Algoritma NH2 yang dibangun oleh penulis ini menggunakan prinsip penyandian Shannon, yaitu *confussion* dan *diffusion*, yang tentunya dapat terlihat bila proses enkripsi dilakukan dengan menggunakan mode *block cipher*. Algoritma NH2 menggunakan proses substitusi pada S-box yang berbeda-beda untuk tiap iterasi pada jaringan feistel, proses pemangkatan modulo, dan proses permut asi pada cipherteks dapat membuat proses dekripsi pesan menjadi lebih susah dilakukan. Berikut akan dijelaskan analisis keamanan yang didasarkan pada serangan-serangan yang memungkinkan dilakukan untuk mendekripsi cipherteks, yaitu dengan menggunakan serangan bruteforce dan serangan analisis frekuensi.

A. Serangan Brute Force

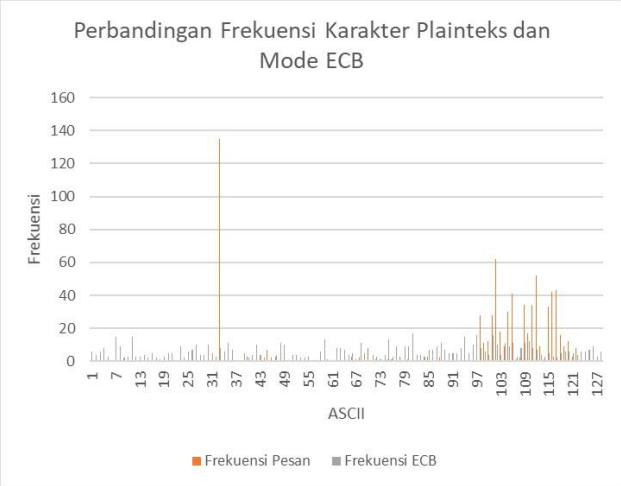
Serangan *brute force* adalah tipe serang yang dilakukan dengan mencari kemungkinan kunci yang digunakan untuk mengenkripsi suatu pesan. Karena metodenya adalah mencoba satu per satu kemungkinan kunci, apabila banyak kemungkinan kunci, waktu yang dibutuhkan untuk menemukan kunci yang tepat akan lama juga.

Panjang kunci yang digunakan dalam algoritma NH2 adalah 32 256 bit. Oleh sebab itu, terdapat 2^{256} = 115.792.089.237.316.195.423.570.985.008.687.907.853.269.984.665.640.564.039.457.584.007.913.129.639.936 kemungkinan kunci. Apabila komputer dapat melakukan

operasi sejumlah 10^8 kali dalam 1 detik, maka waktu yang dibutuhkan untuk menemukan kunci dalam kasus terburuk adalah selama 11.10^{69} tahun atau lebih tepatnya 11.157.920.892.373.161.978.462.380.157.241.004.320.330.798.993.945.869.412.917.119.876.595.712 tahun. Oleh sebab itu, untuk menemukan sebuah kunci, waktu yang dibutuhkan sangat lama dan tidak mungkin dilakukan dengan satu komputer saja. Selain itu, apabila sistem terdistribusi dipakai untuk mencari kuncinya, kemungkinan biaya yang dibutuhkan untuk membangun sistemnya juga tidak sebanding dengan informasinya.

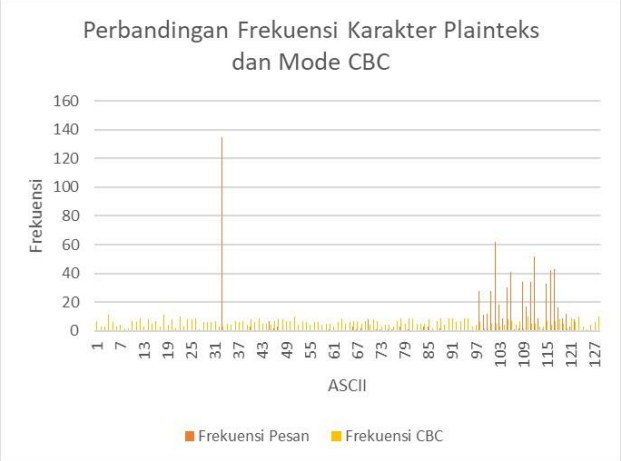
B. Analisis Frekuensi

1) Perbandingan Frekuensi Karakter Plainteks dan ECB



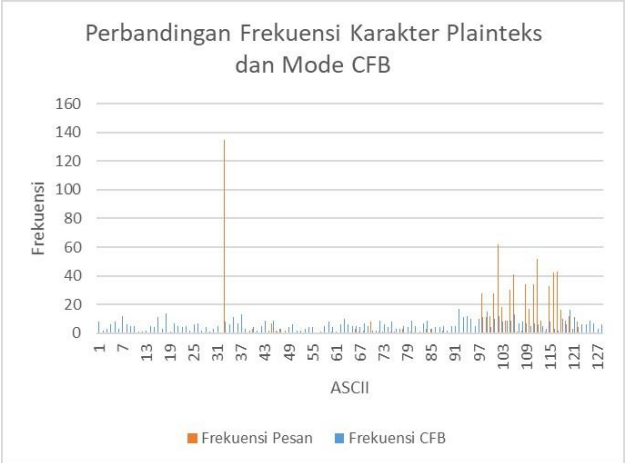
Gambar 4. Grafik perbandingan frekuensi karakter plainteks dan mode ECB

2) Perbandingan Frekuensi Karakter Plainteks dan CBC



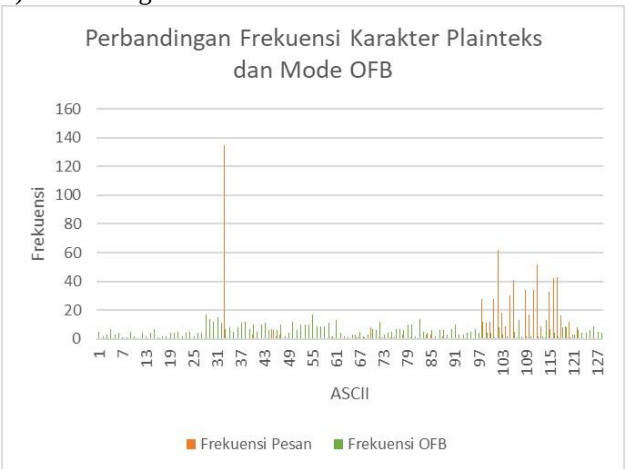
Gambar 5. Grafik perbandingan frekuensi karakter plainteks dan mode CBC

3) Perbandingan Frekuensi Karakter Plainteks dan CFB



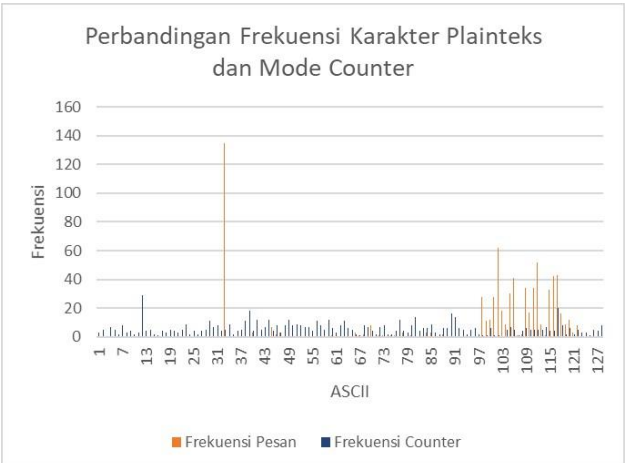
Gambar 6. Grafik perbandingan frekuensi karakter plainteks dan mode CFB

4) Perbandingan Frekuensi Karakter Plainteks dan OFB



Gambar 7. Grafik perbandingan frekuensi karakter plainteks dan mode OFB

5) Perbandingan Frekuensi Karakter Plainteks dan mode Counter



Gambar 8. Grafik perbandingan frekuensi karakter plainteks dan mode *counter*

Pada 5 grafik perbandingan diatas, persebaran frekuensi karakter pada algoritma NH2 ini sudah merata pada berbagai mode percobaan. Dengan adanya persebaran

frekuensi karakter hasil penerapan algoritma NH2 ini, tentunya serangan dengan menggunakan analisis frekuensi untuk mendekripsi teks yang terenkripsi tidak dapat digunakan.

C. Analisis Confusion dan Diffusion

Pada hasil percobaan kami, terlihat bahwa algoritma NH2 ini memiliki sifat *confusion* yang baik. Hal ini terlihat dari tidak adanya korelasi antara karakter tertentu dari kunci dan nilai hexa dari ciphertext. Sebagai contoh, dari hasil percobaan dengan mode ECB, dengan kunci "kriptografi itu seru", karakter kunci ke 3 dan 11 yang berkarakter sama yaitu karakter 'i', tetapi menghasilkan nilai hexa yang berbeda pada hasil enkripsi, yaitu "F4" dan "9B". Fakta tersebut tersebut tentunya akan membuat bingung kriptanalisis dalam menganalisis keterkaitan kunci dengan *ciphertext*.

Prinsip penyandian Shannon diffusion pada algoritma NH2 juga tercapai. Hal ini terlihat dari hasil 5 percobaan yang dilakukan, terlihat bahwa perubahan 1 byte pesan pada plainteks akan menyebabkan nilai blok pesan tertentu atau keseluruhan *ciphertext* akan banyak berubah.

VI. KESIMPULAN DAN SARAN

Algoritma NH2 ini sudah cukup baik untuk mendekripsi suatu pesan karena berdasarkan hasil analisis keamanan yang berdasarkan kemungkinan serangan bruteforce, tentunya proses yang dibutuhkan akan sangat lama, meskipun dilakukan dengan beberapa komputer. Selain itu, teknik serangan analisis frekuensi juga tidak dapat dilakukan untuk algoritma NH2 yang digunakan dalam mode selain EBC karena persebarannya yang cenderung merata. Algoritma NH2 ini juga menerapkan prinsip *confusion* dan *diffusion* dengan adanya modifikasi algoritma, dan melewati beberapa tahap substitusi, rotasi, permutasi, dan operasi matematika. Sebagai saran, algoritma ini dapat dikembangkan dari segi persebaran frekuensi kemunculan karakter pada cipherteks.

VII. UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Tuhan Yang Maha Esa, karena atas bantuan, rahmat, dan berkat-Nya, makalah ini dapat selesai pada waktunya. Tak lupa juga, penulis ingin menyampaikan terima kasih kepada Bapak Dr. Ir. Rinaldi Munir selaku dosen mata kuliah IF4020 Kriptografi yang telah membagikan ilmunya kepada penulis. Selain itu, penulis juga ingin menyampaikan terima kasih kepada kedua orang tua yang selalu mendukung penulis.

REFERENSI

[1] R. Munir, *Diktat Kuliah IF5054 Kriptografi*, Departemen Teknik Informatika Institut Teknologi Bandung, 2005.

[2] R. Munir, *Slide Kuliah IF4020 Kriptografi*, Pengantar Kriptografi, 2018.

[3] R. Munir, *Slide Kuliah IF4020 Kriptografi*, Algoritma Kriptografi Modern, 2018.