DLR

A New Block Cipher Algorithm

Muhammad Helmi Hibatullah - 13520014
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail (gmail): helmihibatullah52@gmail.com

Primanda Adyatma Hafiz - 13520022
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail (gmail): primahafiz@gmail.com

Dimas Faidh Muzaki - 13520156
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail (gmail): dimasfaid@gmail.com

***Abstract*—Pembuatan algoritma cipher blok baru yang disebut DLR terinsipirasi dari algoritma RC6 yang menggunakan jaringan feistel dan fungsi putaran f. Jaringan feistel yang digunakan adalah jaringan feistel tipe dua yang membagi blok menjadi empat bagian. Fungsi putaran f yang dipakai pada jaringan feistel berisi jaringan substitusi dan transposisi. Pembangkitan kunci pada algoritma DLR dibuat seperti jaringan feistel tetapi tidak bersifat reversible. Fungsi pembangkit kunci menerima kunci eksternal sepanjang 128 bit dan menghasilkan 16 kunci 64 bit yang digunakan untuk masing-masing putaran pada jaringan feistel. Berdasarkan hasil percobaan, algoritma DLR sudah memberikan hasil yang baik pada beberapa analisis keamanan seperti prinsip *confusion* dan *diffusion*, pemecahan kunci secara *bruteforce,* dan analisis frekuensi.**

Keywords—algoritma cipher blok baru; jaringan feistel; fungsi putaran; pembangkitan kunci.

I. PENDAHULUAN

Saat ini, banyak terjadi masalah kemanan informasi yang terkait dengan kurang amannya sebuah data dikirimkan sehingga mudah dipecahkan oleh pihak ketiga. Oleh karena itu, diperlukan algoritma kriptografi yang mumpuni untuk menjaga keabsahan, kerahasiaan, kredibilitas, integritas, dan autentikasi data. Adapun kriptograsi dapat dibagi menjadi kriptografi klasik dan kriptografi modern, akan tetapi karena kriptografi klasik sudah dapat mudah dipecahkan maka saat ini kriptografi klasik tidak digunakan lagi sehingga kriptografi modern lebih banyak digunakan. Salah satu cabang dari kriptografi modern yaitu Cipher Block.

Pada *cipher block*, blok plainteks dibagi menjadi blok-blok bit dengan panjang sama. Ukuran blok bit umumnya adalah 64 bit, 128 bit, atau 256 bit. Enkripsi *cipher block* dilakukan pada setiap blok plainteks dengan menggunakan suatu kunci yang dihasilkan dari sebuah kunci eksternal. Akan tetapi, dengan berkembangnya teknologi komputasi saat ini maka perlu didesain *cipher block* yang kuat sehingga tidak dapat dipecahkan dengan mudah baik dengan menggunakan metode *brute force* maupun secara metode analisis. Oleh karena itu, pada makalah ini akan didesain sebuah *cipher block* yang dapat memberikan keamanan data setinggi mungkin dan waktu enkripsi dan dekripsi yang seefisien mungkin.

II. DASAR TEORI

A. Confusion dan Diffusion

Confusion dan *diffusion* merupakan sebuah prinsip yang diperkenalkan untuk mempersulit proses kriptanalisis berbasis statistik. Prinsip ini diperkenalkan oleh Claude Shannon pada tahun 1949 dalam makalah klasiknya, *Communication theory of secrecy systems*. Dua prinsip ini kemudian menjadi panduan dalam merancang sebuah algoritma kriptografi, baik *cipher* maupun *cipher* blok.

1) Confusion

Inti dari prinsip *confusion* adalah menyembunyikan hubungan statistik antara plainteks, cipherteks, dan kunci sehingga tidak mudah untuk mencari korelasi dan menganalisis hubungannya. Prinsip ini dapat direalisasikan dengan mengimplementasikan teknik subtitusi nirlanjar pada algoritma cipher.

2) Diffusion

Prinsip *diffusion* menekankan kepada penyebaran pengaruh satu bit plainteks atau kunci ke sebanyak mungkin bit-bit cipherteks. Dengan begitu, perubahan kecil pada plainteks akan menghasilkan perubahan yang tidak dapat diprediksi pada cipherteks. Prinsip ini dapat terlaksana dengan menggunakan teknik permutasi atau transposisi secara berulang-ulang.

B. Teknik Dasar dalam Kriptografi

1) Teknik Subsitusi

Teknik substitusi menerima input sejumlah m_1 bit atau byte kemudian menghasilkan keluaran berupa m_2 bit atau byte dengan $m_2 \geq m_1$. Teknik ini digunakan untuk membuat data pada setiap blok menjadi tidak berpola sehingga tidak dapat dikenali secara langsung. Teknik substitusi biasanya diimplementasikan dengan menggunakan *S-Box* yang berperan seperti halnya *lookup table* pada *map*.

2) Teknik Permutasi

Teknik permutasi merupakan teknik yang prinsipnya bekerja dengan mengubah susunan bit ke bentuk susunan yang lain. Teknik ini tidak mengubah bit-bit yang ada pada sebuah blok melainkan hanya mengubah susunannya. Teknik permutasi biasanya diimplementasikan dengan menggunakan *P-Box* yang menyimpan data ke mana sebuah data berindeks tertentu perlu dipindahkan.

3) Teknik Transposisi

Teknik transposisi merupakan teknik untuk melakukan pergeseran bit pada suatu data. Pergeseran ini dapat diimplementasikan baik dengan menggunakan *left shift* maupun *right shift* dengan memperlakukan data sebagai kumpulan bit.

C. Mode Operasi Cipher Block

1) Electronic Code Book (ECB)

Mode ECB akan mengenkripsi sebuah blok plainteks P secara individual dan independen dari blok lainnya menjadi sebuah blok cipherteks. Oleh karena itu, pada mode ECB setiap blok plainteks yang sama akan dienkripsi menjadi sebuah cipherteks yang pasti sama pula. Implementasi mode ECB dapat dengan melakukan operasi xor pada setiap blok dengan sebuah key.

2) Cipher Block Chaining (CBC)

Pada mode CBC terdapat ketergantungan antarblok berbeda halnya seperti pada mode ECB. Pada mode ini setiap blok cipherteks bergantung tidak hanya pada blok plainteksnya tetapi juga pada seluruh blok plainteks sebelumnya. Hal ini karena enkripsi pada mode CBC menggunakan hasil cipherteks dari blok sebelumnya.

3) Cipher Feedback (CFB)

Pada dasarnya mode CFB prinsipnya sama dengan mode CBC, akan tetapi mode CFB dapat diterapkan pada pengiriman data yang belum mencapai ukuran satu blok tidak seperti halnya pada mode CBD. Pada mode ini data dienkripsi dalam unit yang lebih kecil daripada ukuran blok misalnya sepanjang 1 bit, 2 bit, dan lain-lain.

4) Output Feedback (OFB)

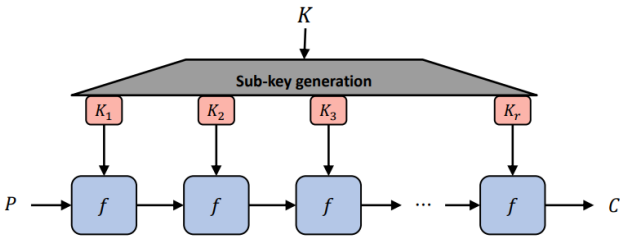
Mode OFB mirip halnya seperti dengan mode CFB tetapi pada mode OFB r-bit hasil enkripsi disalin menjadi elemen paling kanan dari antrian berbeda dengan mode CFB yang menggunakan hasil cipherteks untuk disalin pada posisi paling kanan di antrian.

5) Counter Mode

Mode counter tidak melakukan perantaian sepeti pada CBC tetapi pada mode counter digunakan sebuah nilai counter berupa blok bit yang ukurannya sama dengan ukuran blok plainteks yang diinisialisasi denga sebuah nilai kemudian hasil enkripsinya akan dilakukan operasi xor dengan plainteks pada proses enkripsi. Nilai dari counter akan di-*increment* pada blok-blok berikutnya.

D. Cipher Berulang

Cipher berulang merupakan salah satu algoritma yang bertujuan untuk menghasilkan cipher yang lebih kuat yaitu dengan melakukan *enciphering* sejumlah beberapa kali. Proses *enciphering* sebanyak beberapa kali ini dilakukan dengan berulang kali dilakukan suatu fungsi transformasi f (disebut dengan fungsi putaran) yang mengubah blok plainteks menjadi blok cipherteks. Pada setiap putaran digunakan *subkey* atau kunci putaran yang diguanakan baik dalam enkripsi plainteks maupun dekripsi cipherteks.



Gambar 1. Ilustrasi cipher berulang
Diambil dari
Kriptografi Modern (Bagian 4: Prinsip Perancangan Block Cipher) (itb.ac.id) *pada 4 Maret 2023*

E. Jaringan Feistel

Salah satu implementasi dari struktur *enciphering* pada setiap putaran pada cipher berulang yaitu dengan menggunakan jaringan Feistel. Pada jaringan Feistel blok plainteks dibagi menjadi dua bagian yang akan dilakukan transformasinya masing-masing. Struktur ini bersifat *reversible* karena proses dekripsi dapat dilakukan dengan mengeksekusi jaringan Feistel dari "bawah" ke "atas" atau dengan kata lain dilakukan berkebalikan arah dengan proses enkripsinya. Oleh karena itu, sifat *reversible* ini membuat perancang cipher blok tidak perlu membuat algoritma baru untuk proses dekripsi cipherteks menjadi plainteks.

III. Rancangan Algoritma Cipher

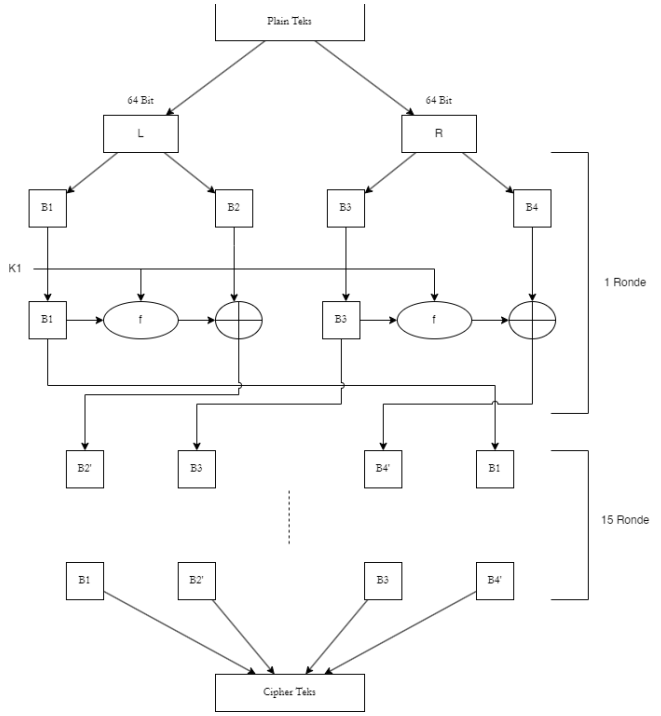
A. Ikhtisar Algoritma

DLR merupakan sebuah algoritma blok cipher yang menggunakan jaringan feistel dan beroperasi dalam satuan bit. Dalam pembuatannya, skema algoritma DLR terinspirasi dari algoritma blok cipher RC6, yaitu menggunakan feistel network dan fungsi putar f. Algortima ini memiliki spesifikasi sebagai berikut:

1) Panjang kunci adalah 128 bit
2) Ukuran blok adalah 128 bit
3) Menggunakan jaringan feistel dengan 16 putaran

B. Skema Jaringan Feistel

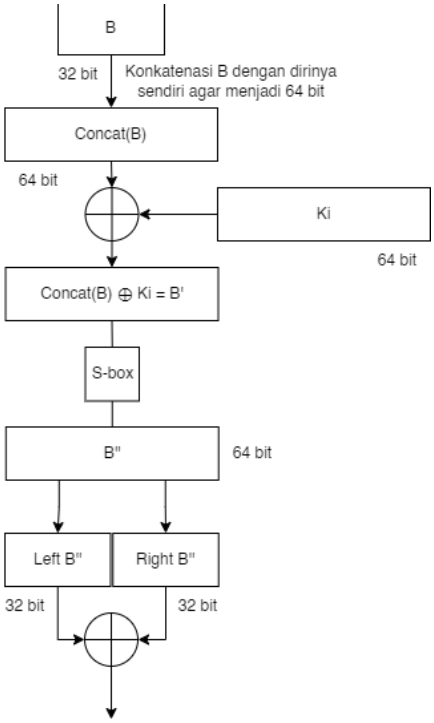
Seperti yang disebutkan sebelumnya, algoritma cipher DLR memanfaatkan jaringan feistel. Jaringan feistel yang digunakan adalah *type-two Feistel*, jenis jaringan feistel yang juga digunakan pada algoritma cipher blok RC6[]. Blok berukuran 128 dibagi menjadi empat bagian yaitu B1, B2, B3, dan B4 yang masing-masing berukuran 32 bit. Pada setiap rondenya, B1 akan diproses menggunakan fungsi putar dan kunci ronde. Hasilnya kemudian akan dilakukan operasi XOR dengan B2 untuk menghasilkan B2'. Begitu juga dengan B3, B3 akan diproses menggunakan fungsi putar dan kunci ronde yang sama digunakan sebelumnya. Hasilnya akan di-XOR kan dengan B4 dan menghasilkan B4'. Untuk ronde selanjutnya, B2' akan digunakan sebagai B1, B3 digunakan sebagai B2, B4' akan digunakan sebagai B3, dan B1 akan digunakan sebagai B4. Proses ini akan diulang-ulang sampai 16 ronde. Di akhir ronde 16, keempat blok berukuran 32 bit akan disatukan kembali untuk membentuk cipherteks. Skema jaringan feistel ini dapat dilihat pada gambar x.



Gambar 2. Skema Jaringan Feistel Algoritma DLR

C. Skema Fungsi Putaran

Fungsi putar yang digunakan pada jaringan feistel berisi jaringan substitusi dan permutasi. Fungsi menerima blok sepanjang 32 bit dan kunci putar sepanjang 64 bit, serta mengembalikan blok yang sudah dienkripsi sepanjang 32 bit. Blok yang masuk kedalam fungsi putar akan dikonkatenasi terlebih dahulu dengan dirinya sendiri agar menjadi 64 bit. Kemudian blok 64 bit tersebut dilakukan operasi XOR dengan kunci putar 64 bit. Setelah itu, blok dilakukan operasi substitusi menggunakan s-box Rijndael seperti pada algoritma cipher AES. Hasilnya kemudian dilakukan pergeseran byte ke kiri sebanyak ronde saat itu pada jaringan feistel. Selanjutnya, bagi blok menjadi dua dengan memotongnya dari tengah, 32 bit awal dan 32 bit akhir. Terakhir, keuda potongan blok tersebut dilakukan operasi XOR sehingga menghasilkan blok baru sepanjang 32 bit. Berikut adalah gambar alur skema fungsi putaran yang digunakan.



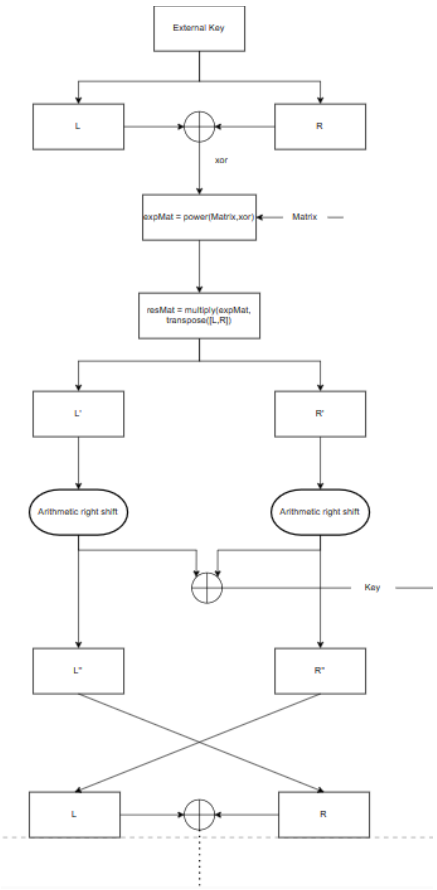
Gambar 3. Skema Fungsi Putar pada Jaringan Feistel Algoritma DLR

D. Skema Pembangkitan Kunci Putaran

Skema pembangkitan kunci putaran menggunakan prinsip yang mirip dengan jaringan Feistel, tetapi skema pembangkitan kunci putaran tidak dibuat *reversible* karena operasi hanya dijalankan secara satu arah saja. Fungsi ini menerima input *external key* berukuran 128 bit dan akan menghasilkan kunci sebanyak 16 buah dengan masing-masing satu kunci untuk setiap putaran dengan ukuran kunci sebesar 64 bit. Untuk membuat kunci yang dibangkitkan seacak mungkin sehingga memenuhi prinsip *confusion* dan *diffusion*, digunakan sebuah konstanta matriks 16x16 yang nantinya akan dipangkatkan lalu dikalikan dengan hasil pemrosesan dari *external key*.

Pertama-tama *external key* akan dipecah menjadi 2 bagian yang memuat 64 bit pertama kunci (L) dan 64 bit terakhir kunci (R). Selanjutnya dilakukan operasi xor pada L dan R dan hasilnya akan menjadi koefisien perpangkatan dari konstanta matriks 16x16. Pada proses perpangkatan matriks digunakan algoritma *fast exponentiation* sehingga perpangkatan matriks dengan koefisien n hanya memerlukan kompleksitas waktu sebesar $O(\log n)$. Kemudian hasil perpangkatan matriks akan dikalikan dengan $transpose([L,R])$ di mana $[L,R]$ akan dipecah per 8 bit (panjang $[L,R] = \frac{128}{8} = 16$) sehingga akan menghasilkan matriks 1x16 yang selanjutnya akan digabungkan menjadi s yang memiliki panjang 128 bit. Lalu s akan dipecah menjadi 2 bagian yang memuat 64 bit pertama kunci (L') dan 64 bit terakhir kunci (R'). Kemudian setiap L' dan R' dilakukan *arithmetical right shift* sebesar jumlah bit yang bernilai 1

menghasilkan L'' dan R''. Kunci yang dihasilkan berupa hasil xor L'' dan R'' yang memiliki panjang 64 bit. Lalu untuk iterasi selanjutnya digunakan $L = R''$ dan $R = L''$ hingga diperoleh kunci sebanyak 16 buah. Skema pembangkitan kunci dapat dilihat pada gambar berikut :



Gambar 4. Skema Pembangkitan Kunci pada Algoritma DLR

IV. Eksperimen dan Pembahsan Hasil

Dalam penulisan makalah ini, eksperimen implementasi algoritma DLR dilakukan dengan menggunakan bahasa pemrograman *python*. Program eksperimen menerima masukan plainteks, cipherteks, dan kunci dalam bentuk *string*. Selain itu, hasil dari cipher akan ditampilkan dalam bentuk *string* unicode. Kami juga memodifikasi program untuk menampilkan heksadesimal untuk kemudahan analisis cipher.

A. *Waktu Enkripsi dan Dekripsi*

Hasil ujicoba cipher ditampilkan pada tabel x untuk plainteks berukuran *small*, tabel x untuk plainteks berukuran medium, tabel x untuk plainteks berukuran large, tabel x untuk plainteks berukuran *very large*.

TABLE I. Hasil Ujicoba 1 DLR Blok Cipher (Small)

Kunci (128 bit)	**Waktu eksekusi**	
	Enkripsi	**Dekripsi**
kriptografiisoke	1.04s	0.8s
Plainteks (128 bit)	**Hasil Cipherteks**	
kamiadatigaorang	å$yÒjçµ½|m6:	
Hasil Heksadesimal Cipherteks		
0xe5 0x24 0x79 0xd2 0x6a 0x9e 0x3b 0x7a 0x8b 0xe7 0xb5 0xbd 0x7c 0x6d 0x36 0x3a		

TABLE II. Hasil Ujicoba 1 DLR Blok Cipher (Medium)

Kunci (128 bit)	**Waktu eksekusi**	
	Enkripsi	**Dekripsi**
kriptografiisoke	5.8s	5.6s
Plainteks (1024 bit)	**Hasil Cipherteks**	
Kenapa kita belajar kriptografi? Supaya kita pandai mengamankan informasi yang sifatnya rahasia agar tidak sembarangan. Paham???	**Program tidak dapat menampilkan string unicode hasil enkripsi**	
Hasil Heksadesimal Cipherteks		
47 D1 4F F9 DA 2A 9D A8 F C7 39 57 21 74 E4 8C 2F AF 47 10 BC D1 20 56 10 4E 71 B 14 D2 8 EB C1 EB 14 E6 D0 52 D5 F7 AD 4A A7 6D 8C 39 37 B0 7E 18 1 3 4 BB 6A 69 D3 CF 81 32 78 86 1E BB 89 36 7D 72 5B 97 AB 41 5E C8 52 D8 FC EA 36 99 40 F6 DD 27 9D C8 C9 3A 68 EC 44 32 E7 68 48 52 8B EF 33 B5 14 D8 9C 9A 28 A7 C7 63 40 A1 16 55 60 CC 4C C FB 9F C1 18 8E FD 22 FB 47 21 70 ED		

TABLE III. Hasil Ujicoba 1 DLR Blok Cipher (Large)

Kunci (128 bit)	**Waktu eksekusi**	
	Enkripsi	**Dekripsi**
kriptografiisoke	72.15s	45.96s
Plainteks (10.240 bit)	**Hasil Cipherteks**	
Dari bangun sampai tidur, ketika berjalan, bekerja, makan, bersantai, hidup kita tak lepas... **< Teks disingkat untuk menghemat halaman makalah >** ...membentuk hidup kita dan bagaimana kita bisa membentuk kebiasaan. Kalau Anda sudah baca buku ini, Anda tak akan lagi melihat dir..	**Program tidak dapat menampilkan string unicode hasil enkripsi**	
Hasil Heksadesimal Cipherteks		
C2 9B C2 9E 11 C3 96 C3 AF C2 B6 4E 30 26 20 C2 9C C3 A4 C2 B9 25 C3 82 11 C3 BB 69 C3 96 43 22 72 58 5A C2 84 C2 BA C2 98 C3 97 13 1E 3B C2 86 C2 A2 10 0B 75 36 1E 09 C2 87 C2 AC C2 BF C2 97 C2 A4 C2 8C 7A 21 C2 97 C3 A4 13 26 59 2C C2 A7 4B 39 C2 A2 C2 84 ..... 81 4F C2 80 C3 9E C2 A9 C3 96 24 55 01 C2 9F 70 C2 9E 6A 31 7D 1A C3 98 C3 A9 C2 A0 C2 AF 7C 12 C3 89 C2 BA 26 C3 95 C3 87 C2 81 C2 99 64 6E 23 C2 95 65 C2 AA 20 C2 AB 7F C3 07 27 C2 8A 27 68 C3 B4 2D C2 83 C3 B2 0B 6F C3 8E C2 8B C2 B1 C2 9D 2B 19 27 C3 A5 0E C3 94 33 1C 74 4B 60 C3 BC 54 C2 9D C3 8B C3 BB 36 68 4F C3 88 52 C3 AF C3 8C C3 94 44 42 C3 9A C2 80 C3 94 C3 B2 00 23 29 08 1E C3 B8 02 C2 9A 50 C3 85 C2 96 06 38 23 75 4F 3A 68 48 22 74		

TABLE IV. Hasil Ujicoba 1 DLR Blok Cipher (Very Large)

Kunci (128 bit)	**Waktu eksekusi**	
	Enkripsi	**Dekripsi**
kriptografiisoke	5 menit 8 detik	4 menit 41 detik
Plainteks (10.240 bit)	**Hasil Cipherteks**	
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut	**Program tidak dapat menampilkan string unicode hasil enkripsi**	

eget nulla lacus. Integer condimentum id nunc at vulputate. Ut vehicula laoreet libero ac pellentesque. Phasellus .....faucibus felis nunc sed ligula. Maecenas efficitur justo vitae maximus mollis. Etiam quis sollicitudin libero. Vestibulum elementum mi sed dui posuere egestas. Integer auctor gravida justo.	
Hasil Heksadesimal Cipherteks	
03 C3 A2 C2 A6 45 17 69 20 0E C3 91 70 4A C3 83 42 13 53 47 C2 89 C3 AD 2D 09 C3 AB C3 B1 39 47 C2 8E 6D 60 2B C3 A8 5F 32 C2 B3 05 C3 A6 C3 93 C2 86 C3 BC C3 9C 57 70 C3 AD ..... 08 4D 52 71 C2 82 C3 AB 35 C3 86 C3 91 C2 A2 00 C3 8C C2 B3 54 17 0B C2 AF C2 8F 79 24 6B 59 49 C2 8D 7B C2 AF C2 A3 C2 93 C3 B6 C2 B0 C3 A9 C3 97 57 C3 96 5F C2 A8 C3 95 7B C3 A4 0C 15 49 65 C2 85 C3 A0 C3 BF C3 95 52 13 34 C3 86 C2 AA 39 4B C2 A0 C2 9E 39 C3 84 C2 9C 69 1E 3A 05 6A C2 B6 C3 BE 3E 2E 64 C2 82 C2 BE 1E 7D 29 C2 88 1E 50 C3 B3 C2 A0 C2 9F 4B	

B. *Analisis Keamanan*

1) *Analisis Efek longsoran (avalanche effect)*

Efek longsoran dapat dijadikan penanda bahwa sebuah algoritma cipher telah berhasil melaksanakan prinsip diffusion. Berdasarkan ujicoba yang dilakukan dan ditampilkan pada tabel x, algoritma yang kami rancang memiliki efek longsoran untuk perubahan plainteks maupun perubahan kunci. Efek longsoran akibat perubahan plainteks tidaklah begitu besar, perubahan 1 byte pada plainteks hanya mengubah 8 byte saja dan membentuk sebuah pola tertentu. Sementara itu, perubahan kunci dapat memberikan efek longsoran yang besar, yaitu perubahan seluruh cipherteks dalam satuan byte.

TABLE V. UJICOBA EFEK LONGSORAN

Kunci	Plainteks	Cipherteks
kriptografiisoke	kami**i**adatigaorang	0xe5 0x24 0x79 0xd2 0x6a 0x9e 0x3b 0x7a 0x8b 0xe7 0xb5 0xbd 0x7c 0x6d 0x36 0x3a
	kam**u**adatigaorang	0xe5 **0x69** 0x79 **0xdc** 0x6a **0x6f** 0x3b **0x45** **0xbf** 0xe7 **0xd1** 0xbd **0x1f** 0x6d **0xaf** 0x3a
kriptografiisok**e**	kamiadatigaorang	0xe5 0x24 0x79 0xd2 0x6a 0x9e 0x3b 0x7a 0x8b 0xe7 0xb5 0xbd 0x7c 0x6d 0x36 0x3a
kriptografiisok**a**		0x78 0x38 0x5a 0xd 0x6 0xfa 0x87 0xae 0x1b 0x39 0xb4 0x2c 0x24 0xd0 0xbe 0x60

Perlu diingat bahwa mode operasi algoritma ini adalah ECB, yaitu setiap blok dienkripsi secara individual dan independen dari blok lainnya. Dengan begitu, efek longsoran tidak terjadi di tingkat interblok, tetapi hanya di dalam blok saja.

2) *Analisis Ruang Kunci*

Cipher ini menggunakan kunci sepanjang 128-bit atau 16 byte dimana untuk penyerangan dengan metode *brute force* akan dicari kombinasi kunci dengan besar ruang pencarian adalah sebesar

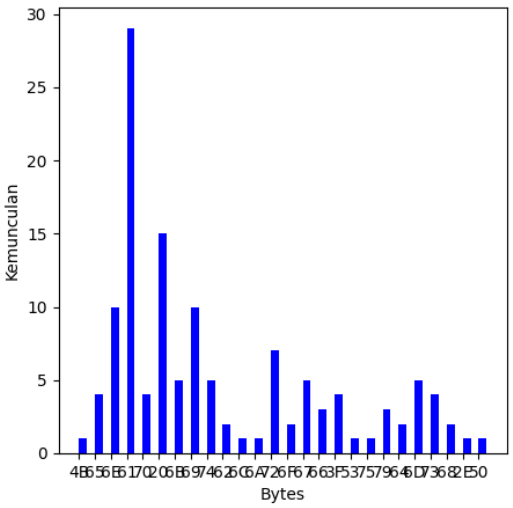
$$2^{128} = 3{,}4 \,.\, 10^{38}$$

Dengan ruang pencarian sebesar itu, jika kita memiliki computer tercepat yang dapat mencoba 1 juta kemungkinan dalam 1 detik, maka akan dibutuhkan 5,4 x 1024 tahun untuk dapat menemukan kunci yang tepat.

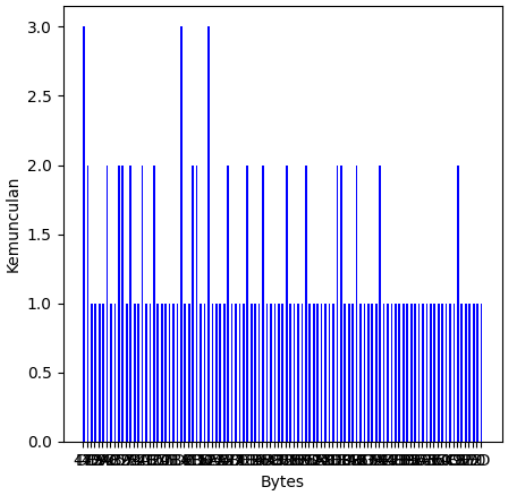
3) *Analisis Statistik*

Analisis statistik dilakukan untuk menguji keberhasilan cipher dalam menerapkan prinsip confusion. Analisis ini dilakukan dengan cara menghitung frekuensi kemunculan byte dari suatu plainteks dengan frekuensi kemunculan byte pada cipherteksnya. Untuk untuk analisis statistik, kita akan menggunakan ujicoba plainteks medium seperti pada poin 1.

Hasil analisis kemunculan byte cukup baik. Plainteks yang memiliki 26 byte unik berhasil disebar oleh cipher menjadi 103 byte unik. Dapat dilihat pada grafik di gambar x bahwa pembagiannya pun cukup merata, yaitu banyak byte yang hanya muncul sekali.



Gambar 5. Grafik Frekuensi Kemunculan Byte pada Plainteks Medium



Gambar 6. Grafik Frekuensi Kemunculan Byte pada Cipherteks Medium

V. KESIMPULAN DAN SARAN

Algoritma DLR sudah memberikan hasil yang baik pada beberapa analisis keamanan seperti prinsip *confusion* dan *diffusion*, pemecahan kunci secara *bruteforce*, dan analisis frekuensi. Hal ini dapat tercapai karena dalam fungsi putar f menggunakan teknik subtitusi dan permutasi. Selain itu, dilakukan juga *enciphering* berulang melalui jaringan feistel.

Walaupun begitu, kami menyadari bahwa cipher ini memiliki kekurangan dalam beberapa aspek sehingga dapat diperbaiki kedepannya. Berdasarkan ujicoba, cipher ini membutuhkan waktu yang cukup lama untuk melalukan enkripsi dan dekripsi. Selain itu, mode operasi yang bersifat ECB membuat algoritma ini rentan kriptanalisis apabila terdapat blok-blok 128 bit yang sama persis. Oleh sebab itu, pengembangan lebih lanjut dapat dilakukan dengan mengubah mode operasi menjadi CBC, CFB, OFB, maupun CTR. Dengan begitu, efek longsoran (*avalanche effect)* meningkat ke level blok.

REFERENSI

https://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2022-2023/13-Block-Cipher-Bagian1-2023.pdf Diakses pada 4 Maret 2023

https://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2022-2023/14-Prancangan-block-cipher-2023.pdf Diakses pada 4 Maret 2023