Algoritma Brick Cipher

Bimo Adityarahman Wiraputra[1], **Ricky Yuliawan**[2].

[1,2] Program Studi Teknik Informatika, Sekolah Teknik Elektro dan Informatika (STEI), Institut Teknologi Bandung (ITB), Jalan Ganesha 10, Bandung 40132
E-mail: 13517004@std.stei.itb.ac.id, 13517025@std.stei.itb.ac.id

Abstraksi. Block Cipher merupakan salah satu teknik kriptografi modern yang populer digunakan. Terlebih lagi karena teknologi komputer berkembang dengan sangat cepat dan terdapat kebutuhan yang besar untuk layanan kriptografi. Sekarang sudah banyak sekali rancangan serta variasi dari algoritma Block Cipher. Pada makalah ini diusulkan rancangan baru dari algoritma block cipher, yaitu Brick Cipher. Brick Cipher menggunakan prinsip *Confusion* dan *Diffusion*. Kemudian algoritma ini menggunakan jaringan Feistel untuk melakukan fungsi substitusi, fungsi permutasi, dan fungsi pengacakan yang memanfaatkan pembangkitan kunci putaran. Algoritma ini menghasilkan hasil enkripsi yang cukup aman dan cocok untuk digunakan sebagai pondasi untuk pengembangan algoritma kriptografi ke depannya. Makalah ini akan menjelaskan bagaimana perancangan algoritma Brick Cipher yang penulis buat. **Keywords**: Kriptografi, Block Cipher, Variasi Block Cipher, *Confusion*, *Diffusion*, Jaringan Feistel, *round key*

1. Latar Belakang

Seiring perkembangan zaman, hal yang juga sangat terasa perkembangannya adalah teknologi informasi, khususnya kebutuhan akan pertukaran informasi yang semakin hari semakin bertambah. Namun, pertukaran informasi ini harusnya memiliki keamanan akan data yang ada di dalamnya. Pengamanan tersebut menggunakan kriptografi. Secara etimologi, kata kriptografi (*Cryptography*) berasal dari bahasa Yunani, yaitu *'kryptos'* yang artinya 'yang tersembunyi' (*hidden/secret*) dan *'graphein'* yang artinya 'tulisan' (*write*) (Prayudi, 2005). Kriptografi atau biasa disebut dengan sandisastra merupakan ilmu tentang teknik matematis yang digunakan melakukan enkripsi dimana *plaintext* atau teks asli diacak menggunakan suatu kunci enkripsi menjadi *ciphertext*, yaitu teks acak yang sulit dibaca oleh seseorang yang tidak memiliki kunci dekripsi. *Ciphertext* ini hanya dapat kembali ke *plaintext* dengan melakukan dekripsi menggunakan kunci dekripsinya, sedangkan peluang mendapat kembali *plaintext* oleh seseorang yang tidak mempunyai kunci dekripsinya dalam waktu yang tidak terlalu lama adalah sangat kecil. Kriptografi sangat mudah dijumpai meskipun mungkin hanya sebagian kecil dari masyarakat di dunia yang merasakan langsung kegunaan dari kriptografi tersebut.

Kriptografi dibagi menjadi dua, yaitu kriptografi klasik dan kriptografi modern. Kriptografi klasik merupakan teknik kriptografi yang sudah digunakan pada zaman dahulu sebelum komputer ditemukan atau sudah ditemukan namun belum secanggih sekarang yang caranya hanya melakukan pengacakan plaintext atau hanya melakukan pengacakan pada huruf A - Z. Salah satu contohnya adalah Caesar cipher, Vigènere cipher, Playfair cipher, dan lain-lainnya. Sedangkan, kriptografi modern merupakan teknik kriptografi yang beroperasi dalam mode bit ketimbang mode karakter dan pengoprasi kriptografi ini dalam mode bit berarti semua data dan informasi (*plaintext*, kunci, dan *ciphertext*) dinyatakan dalam rangkaian string ataupun bit biner 0 dan 1. Teknik enkripsi dan dekripsinya

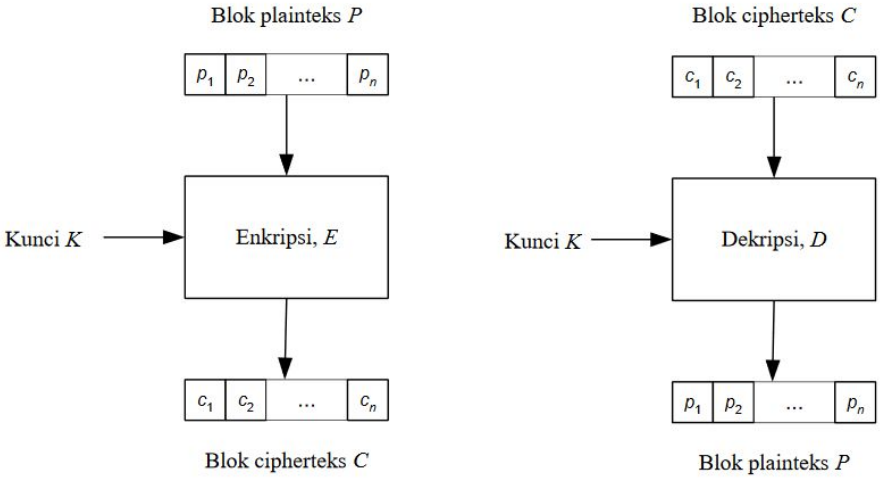
memproses semua data dan informasi dalam bentuk rangkaian bit dan kemudian rangkaian bit yang menyatakan plaintext dienkripsi menjadi ciphertext dalam bentuk rangkaian bit, demikian sebaliknya. Salah satu contohnya adalah AES, RSA, 3DES, DES, dan algoritma lainnya. Kriptografi modern masih menggunakan prinsip-prinsip dari kriptografi klasik yang digabungkan, hanya saja kriptografi modern sering beroperasi dalam bentuk bit dan operasi yang paling banyak digunakannya adalah operasi XOR. Membangun kriptografi modern yang baik tergolong sulit karena studi untuk memecahkan suatu algoritma kriptografi sudah cukup banyak berkembang.

Pada makalah ini akan dibahas block cipher baru yang dirancang oleh penulis. Algoritma ini penulis namakan Brick Cipher. Alasan kami menamainya Brick cipher karena algoritma block cipher ini berasal dari akronim dua penulis dan diharapkan dapat menjadi pondasi yang kuat untuk kriptografi modern kedepannya.

2. Dasar Teori

2.1. *Block Cipher*

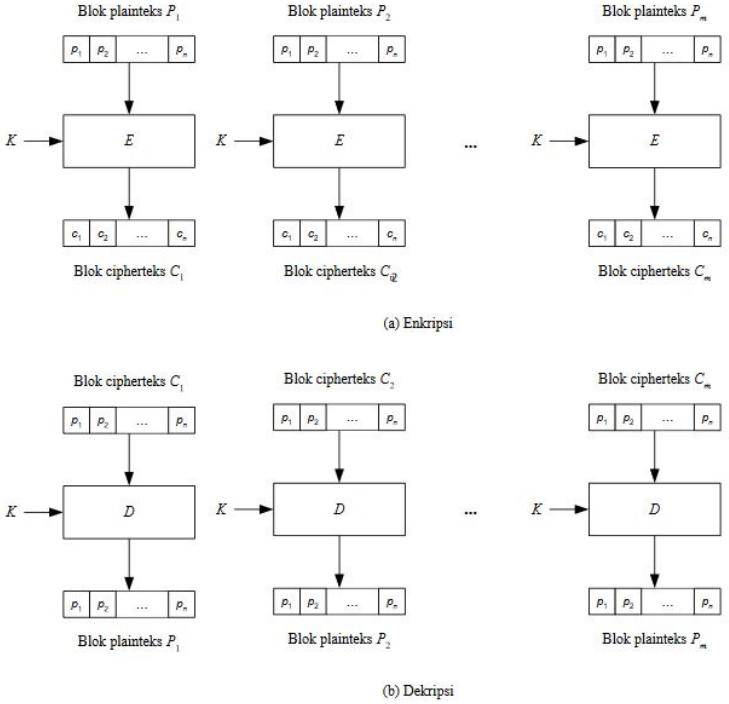
Salah satu contoh kriptografi modern adalah *block cipher*. *Block cipher* merupakan algoritma deterministik pada kriptografi yang bekerja pada sekelompok bit berukuran tetap yang disebut blok. *Block cipher* menggunakan kunci simetris dan melakukan proses enkripsi dan dekripsi pada blok bit dengan ukuran tertentu. Bit-bit *plaintext* pada blok cipher dibagi menjadi blok-blok bit dengan panjang sama, misalnya 64-bit. Panjang blok pada *ciphertext* sama dengan panjang blok pada *plaintext*. Enkripsi dilakukan terhadap blok *plaintext* dengan bit-bit kunci dengan panjang kunci eksternalnya tidak harus sama dengan panjang block pada *plaintext*.[1]



Gambar 1. Skema Enkripsi dan Dekripsi pada Block Cipher
(Sumber:http://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2020-2021/Kripto-modern-2020-bagian3.pdf)

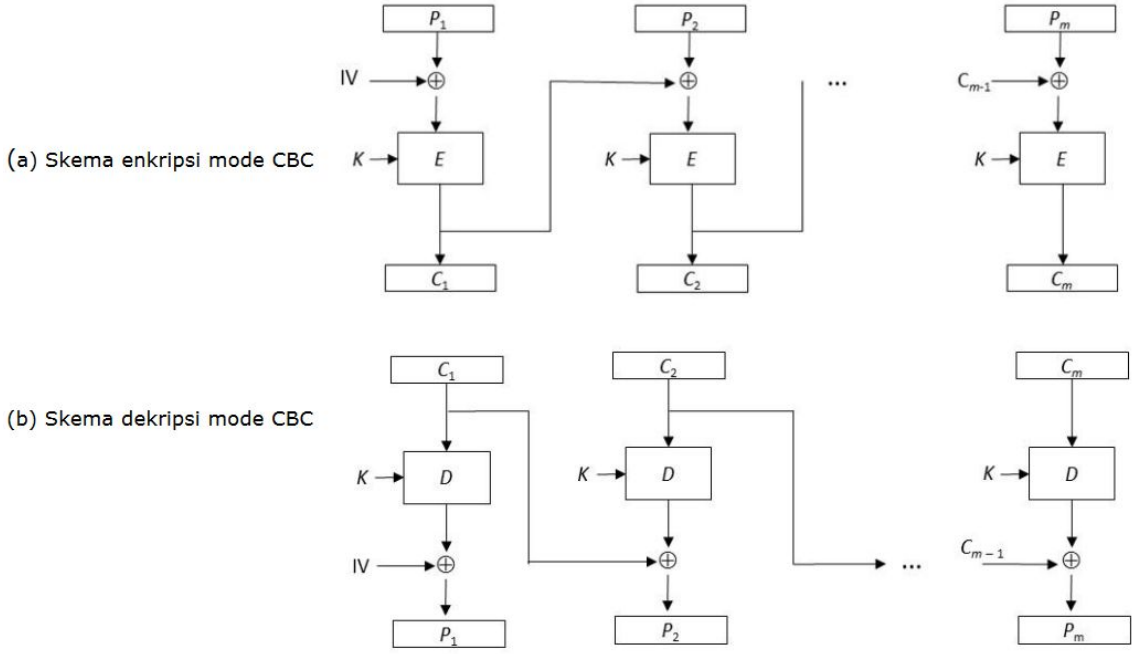
Block cipher memiliki beberapa mode operasi yang berkaitan dengan cara blok dioperasikan sebelum dienkripsi atau didekripsi oleh fungsi E dan D pada masing-masing blok, diantaranya:

1) *Electronic Code Book* (*ECB*)
   Mode *Electronic Code Book* (*ECB*) merupakan mode enkripsi paling sederhana yang dinamai dari buku kode fisik. Pesan-pesan dibagi ke dalam blok-blok lalu dienkripsi terpisah dan tidak mempengaruhi satu sama lain. Karena setiap blok *plaintext* dienkripsi secara independen, maka tidak perlu mengenkripsi pesan secara sekuensial/linier. Kesalahan satu atau lebih bit pada block ciphertext hanya mempengaruhi ciphertext yang bersangkutan pada proses dekripsi. Namun, karena plaintext sering mengandung bagian yang berulang (sehingga terdapat blok-blok plaintext yang sama), maka enkripsinya menghasilkan block ciphertext yang sama pula. [1]



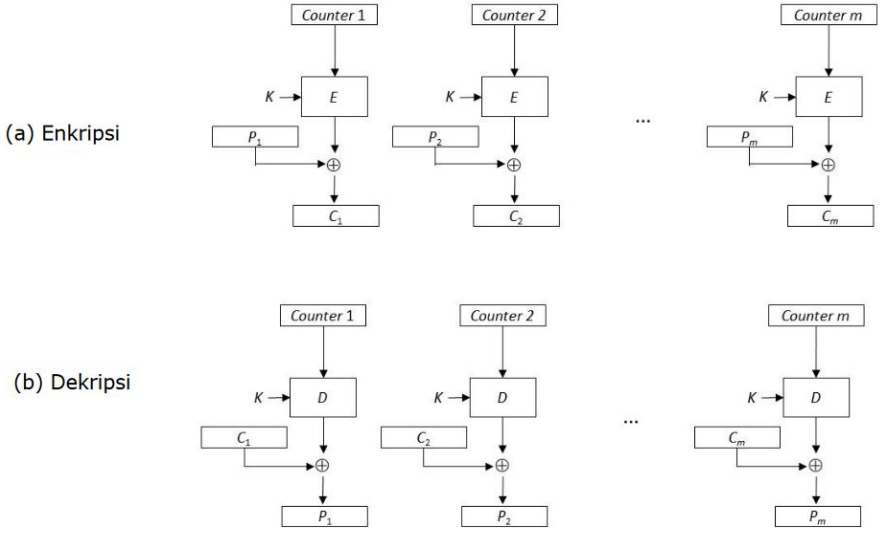
Gambar 2. Skema Enkripsi dan Dekripsi pada ECB
(Sumber:http://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2020-2021/Kripto-modern-2020-bagian3.pdf)

2) Cipher Block Chaining (CBC)
Dalam mode Cipher Block Chaining (CBC), tiap blok *plaintext* di-XOR dengan *ciphertext* sebelumnya sebelum dienkripsi. Dengan cara ini, tiap blok *ciphertext* bergantung pada semua blok *plaintext* yang telah diproses hingga blok saat itu. Untuk membuat tiap pesan unik, vektor inisialisasi harus dipakai dalam blok pertama. Kelebihan dari mode CBC adalah blok-blok plaintext yang sama tidak selalu menghasilkan blok-blok ciphertext yang sama. Oleh karena blok-blok plaintext yang sama tidak menghasilkan blok-blok ciphertext yang sama, maka kriptanalisis menjadi lebih sulit. Namun, kesalahan satu bit pada sebuah blok plaintext akan menghasilkan kesalahan pada blok ciphertext yang berkoresponden dan kesalahan tersebut merambat ke semua blok ciphertext berikutnya.[1]



Gambar 3. Skema Enkripsi dan Dekripsi pada CBC
(Sumber:http://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2020-2021/Kripto-modern-2020-bagian3.pdf)

3) Counter Mode (CTR)
Mode Counter (CTR) juga dikenal sebagai *integer counter mode* (ICM) dan *segmented integer counter* (SIC). Seperti OFB, mode counter mengubah block cipher menjadi stream cipher. Mode Counter membangkitkan blok aliran kunci dengan mengenkripsi nilai counter selanjutnya. Namun, mode counter tidak melakukan perantaian (*chaining*) seperti pada CBC. Nilai counter harus berbeda dari setiap blok yang dienkripsi. Pada mulanya, untuk enkripsi blok pertama, counter diinisialisasi dengan sebuah nilai. Selanjutnya, untuk enkripsi blok-blok berikutnya counter dinaikkan nilainya satu.



Gambar 4. Skema Enkripsi dan Dekripsi pada Counter Mode
(Sumber:http://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2020-2021/Kripto-modern-2020-bagian3.pdf)

2.2. Prinsip *Diffusion* dan *Confusion*

Claude Shannon dalam makalah klasiknya tahun 1949, *Communication Theory of Secrecy Systems*, memperkenalkan prinsip *confusion* dan *diffusion* untuk membuat serangan statistik menjadi rumit. Dua prinsip tersebut menjadi panduan dalam merancang algoritma kriptografi.[2]

1) Prinsip *diffusion*
   Prinsip *diffusion* merupakan prinsip yang menyebarkan pengaruh satu bit *plaintext* atau kunci ke sebanyak mungkin *ciphertext*. Secara umum, perubahan pada satu bit di *plaintext* harusnya mengubah sekitar setengah dari hasil *ciphertext* dan juga kebalikannya. *Confusion* dapat direalisasikan dengan menggunakan operasi permutasi. Mode *block cipher* yang menggunakan prinsip tersebut adalah CBC dan CFB.

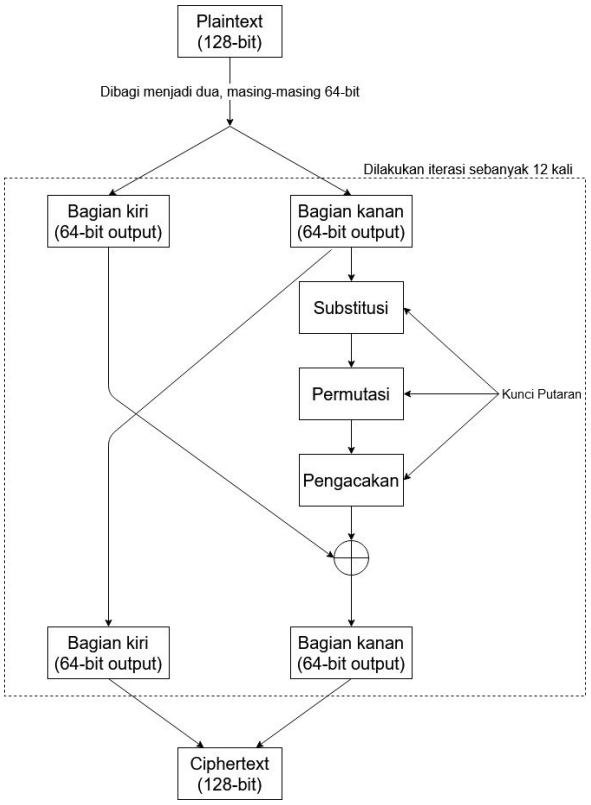
2) Prinsip *confusion*
   Prinsip *confusion* merupakan prinsip yang menyembunyikan hubungan apapun yang ada antara *plaintext*, *ciphertext*, dan kunci. Prinsip confusion membuat kriptanalis frustasi untuk mencari pola-pola statistik yang muncul pada *ciphertext*. Secara umum, perubahan pada satu bit di kunci sebaiknya mengubah perhitungan di hampir semua hasil *ciphertext*. Salah satu contohnya adalah *One-Time Pad*. Confusion dapat direalisasikan dengan menggunakan algoritma substitusi yang kompleks.

2.3. Jaringan Feistel

Jaringan Feistel merupakan struktur sistem yang dipakai dalam penyusunan penyandian blok. Jaringan Feistel banyak dipakai pada algoritma kriptografi, seperti DES, karena model ini bersifat *reversible* untuk proses enkripsi dan dekripsi. Sifat *reversible* inilah yang membuatnya tidak perlu membuat algoritma baru untuk mendekripsi *ciphertext* menjadi *plaintext*. Dalam jaringan Feistel, blok yang hendak dienkripsi dibagi menjadi dua bagian, lalu dilakukan beberapa putaran dimana di setiap putaran, blok pertama disubstitusi dengan blok kedua dan blok kedua disubstitusi dengan blok pertama yang dixorkan dengan fungsi putaran dari blok kedua. Fungsi putaran ini tidak diwajibkan untuk bersifat *invertible* untuk dapat melakukan dekripsi di jaringan Feistel, sehingga lebih mudah untuk dikembangkan.

3. Desain Algoritma Brick Cipher

Brick cipher merupakan algoritma *block cipher* yang beroperasi pada blok berukuran 128 bit. Kriptografi ini menerima kunci sepanjang 128 bit juga. Enkripsi pada masing-masing blok menggunakan jaringan Feistel dengan dua belas putaran. Dibangkitkan dua belas kunci putaran sepanjang 64 bit dari kunci kriptografi tersebut. Fungsi putaran yang digunakan terdiri dari tiga tahap, yaitu tahap substitusi, permutasi, dan pengacakan. Menggunakan fungsi enkripsi blok ini, kriptografi dapat dijalankan dalam mode ECB, CBC, dan CTR berdasarkan cara kerja yang sudah dijelaskan sebelumnya.



Gambar 5. Struktur Algoritma Brick Cipher

Penjelasan masing-masing bagian dari algoritma yaitu:

1) Pembangkitan kunci putaran

   Pembangkitan kunci putaran terinspirasi dari algoritma pembangkit bilangan acak ringan bernama *middle square method* menggunakan barisan Weyl.[3] Algoritma ini dikenal sebagai salah satu algoritma tercepat yang berhasil melewati rangkaian tes statistik untuk pembangkit bilangan acak seperti BigCrush dan PractRand. Untuk membangkitkan barisan sepanjang n bit, dari hasil bilangan sebelumnya dikuadratkan dan dijumlahkan dengan barisan Weyl, yang merupakan kelipatan dari suatu bilangan besar acak. Operasi ini menghasilkan bilangan sepanjang 2n bit yang diambil n bit tengahnya sebagai hasil dari pembangkitan ini.

   Kunci 128 bit diubah terlebih dahulu menjadi 64 bit dengan melakukan xor pada rentang kunci awal. Lalu dilakukan pembangkitan *middle square method* menggunakan barisan Weyl dilakukan dengan nilai n sebesar 64, dimana kunci awal sepanjang 128 bit digunakan sebagai nilai mula-mula pada pembangkit dan barisan Weyl yang digunakan adalah kelipatan dari 7295995950422153645567912886726701447l.

2) Fungsi substitusi

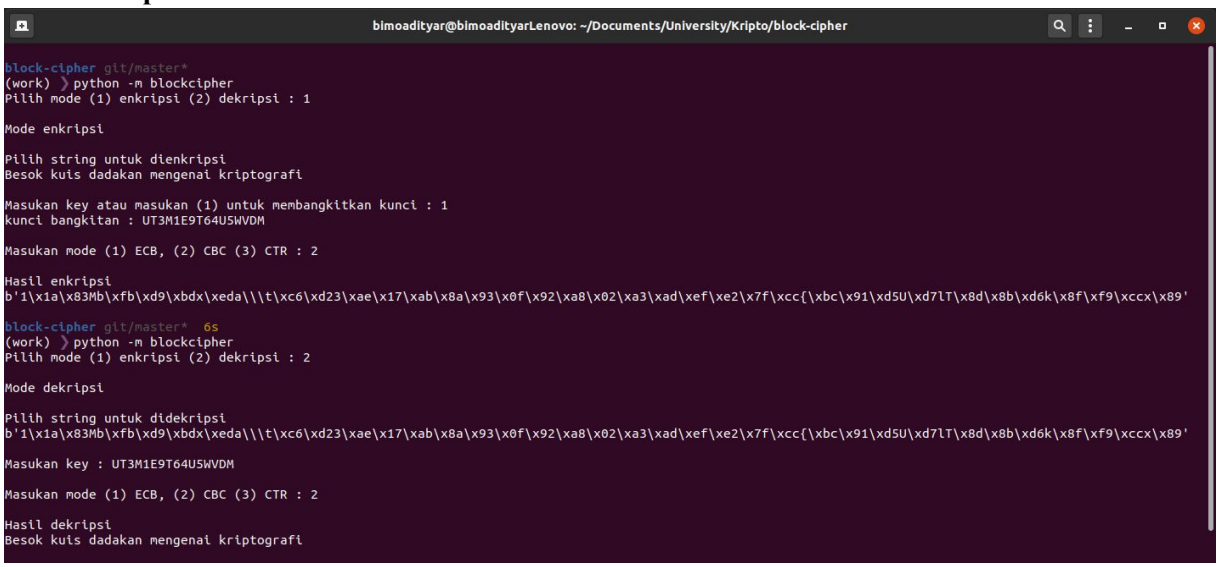
   Fungsi substitusi menggunakan tabel Sbox yang berasal dari transformasi linear transformasional di medan Galois seperti pada AES yang dipermutasi untuk menambah faktor acak berdasarkan.[4] Selain hanya menggunakan Sbox, pemilihan kotak hasil substitusi berdasarkan pada byte yang ditransformasi dan byte pada kunci putaran yang digilir. Kedua komponen empat bit pertama dan kedua di masing-masing byte tersebut dijumlahkan dan dimodulo 16 untuk mendapatkan indeks pada Sbox.

3) Fungsi permutasi

   Permutasi dilakukan pada 8 byte yang ada menggunakan pemetaan yang dibangkitkan secara acak dan memiliki banyak *cycle* satu, yaitu [5, 7, 6, 3, 1, 4, 2, 8].

4) Fungsi pengacakan
Dilakukan pencampuran antara byte di potongan dengan melakukan penjumlahan antara byte yang bersebelahan. Dalam empat iterasi, untuk setiap byte akan dijumlahkan dengan byte sebelumnya modulo 256. Penjumlahan hanya akan dilakukan apabila jumlah byte yang akan dijumlahkan dan byte di kunci putaran yang digilir tidak bersisa satu module tiga untuk menghindari linearitas dari fungsi.

4. Eksperimen dan Analisis Hasil



Gambar 6. Hasil eksperimen kode

Kode diimplementasi di platform Python. Karena kode melakukan penambahan bit agar teks dapat dibagi menjadi kumpulan blok, ukuran hasil enkripsi merupakan pembulatan ke atas dari ukuran *plaintext* sehingga menjadi kelipatan dari 256 bit atau 16 byte.

4.1. Eksperimen

Tabel 1. Tabel kecepatan enkripsi (masing-masing 5 percobaan)

Jenis	Kecepatan rata-rata enkripsi (s)	Kecepatan rata-rata dekripsi (s)
ECB-100byte	0.0010435581	0.0010582447
ECB-10000byte	0.0848991871	0.0845222473
ECB-1000000byte	8.6019402027	8.5402530670
CBC-100byte	0.0010917187	0.0010522842
CBC-10000byte	0.0852603912	0.0850361347
CBC-1000000byte	8.6020801067	8.5910693645
CTR-100byte	0.0011602879	0.0011068821
CTR-10000byte	0.0866822720	0.0858194828
CTR-1000000byte	8.7485237122	8.7663721085

Dapat dilihat dari tabel 1 bahwa kecepatan enkripsi kurang lebih sama untuk enkripsi dan dekripsi maupun untuk mode ECB, CBC, dan CTR. Kecepatan algoritma kurang lebih linear terhadap ukuran masukan, dengan waktu pemrosesan sekitar 8,5 mikrodetik per byte.

Tabel 2. Tabel banyak bit yang berubah di ciphertext saat satu bit acak pada kunci diganti

Jenis	Bit total pada *ciphertext*	Banyak bit yang berubah di *ciphertext*
ECB-100byte	896	446.0
ECB-10000byte	80128	40049.6
ECB-1000000byte	8000128	4000387.6
CBC-100byte	896	452.4
CBC-10000byte	80128	40135.8
CBC-1000000byte	8000128	3999519.6
CTR-100byte	896	440.8
CTR-10000byte	80128	40077.6
CTR-1000000byte	8000128	3999650.2

Dapat dilihat dari tabel 2 bahwa banyak bit yang berubah terhadap perubahan pada kunci enkripsi sekitar setengah dari semua bit yang ada. Ini menunjukkan bahwa algoritma mengikuti prinsip *confusion* Shannon.

Tabel 3. Tabel banyak bit yang berubah di *ciphertext* saat satu bit acak pada *plaintext* diganti

Jenis	Bit total pada *ciphertext*	Banyak bit yang berubah di *ciphertext*
ECB-100byte	896	62.4
ECB-10000byte	80128	68.0
ECB-1000000byte	8000128	62.2
CBC-100byte	896	217.4
CBC-10000byte	80128	15245.4
CBC-1000000byte	8000128	2560580.2
CTR-100byte	896	1.0
CTR-10000byte	80128	1.0
CTR-1000000byte	8000128	1.0

Dapat dilihat dari tabel 3 bahwa banyak bit yang berubah terhadap perubahan pada *plaintext* tidak melebihi dari panjang blok sesuai dengan perilaku dari mode ECB. Di mode CTR, perubahan hanya mempengaruhi di posisi yang sama dengan bit yang berubah. Sedangkan di mode CBC, perubahan

dapat menjalar ke blok berikutnya, sehingga banyak perubahan dipengaruhi oleh di blok berapa bit yang berubah tersebut. Secara umum, terjadi perubahan dengan rasio setengah dari bagian yang dapat dipengaruhi perubahan tersebut, sehingga algoritma memenuhi prinsip *diffusion* Shannon.

Tabel 4. Tabel banyak bit yang berubah di *plaintext* saat satu bit acak pada *ciphertext* diganti

Jenis	Bit total pada *plaintext*	Banyak bit yang berubah di *plaintext*
ECB-100byte	800	48.6
ECB-10000byte	80000	62.8
ECB-1000000byte	8000000	62.2
CBC-100byte	800	68.0
CBC-10000byte	80000	65.4
CBC-1000000byte	8000000	66.8
CTR-100byte	800	1.0
CTR-10000byte	80000	1.0
CTR-1000000byte	8000000	1.0

Dapat dilihat dari tabel 4 bahwa banyak bit yang berubah terhadap perubahan bit di *ciphertext* pada *plaintext* bernilai sekitar setengah blok untuk ECB dan CBC dan hanya satu pada mode CTR. Secara umum, terjadi perubahan dengan rasio setengah dari bagian yang dapat dipengaruhi perubahan tersebut, sehingga ini juga menunjukkan algoritma ini memenuhi prinsip *diffusion* Shannon.

4.2. Analisis Keamanan

Keamanan block cipher terhadap serangan *brute force* bergantung pada jumlah kemungkinan kunci yang mungkin digunakan. Kunci yang digunakan pada Brick Cipher yang kami rancang terdiri dari 128-bit nilai biner yang menghasilkan 2^{128} kemungkinan kunci. Jika percobaan *brute force* menggunakan mesin yang memiliki kemampuan melakukan satu juta operasi per detik, maka waktu yang dibutuhkan untuk mencoba semua kemungkinan adalah sekitar 3.4028237×10^{32} detik atau sekitar 1.0790283×10^{25} tahun. Ataupun jika percobaan *brute force* menggunakan mesin yang memiliki kemampuan melakukan satu triliun operasi per detik, maka waktu yang dibutuhkan untuk mencoba semua kemungkinan adalah sekitar 3.4028237×10^{26} detik atau sekitar 1.0790283×10^{19} tahun. Maka, algoritma Brick Cipher cukup aman dari serangan *brute force*.

Berdasarkan pada hasil eksperimen di atas, prinsip *confusion* dan *diffusion* dari Shannon pun berhasil terpenuhi. Untuk prinsip *confusion* dapat dibuktikan bahwa dari algoritma Brick Cipher, banyak bit yang berubah terhadap perubahan pada kunci enkripsi sekitar setengah dari semua bit yang ada. Untuk prinsip *diffusion*-nya dapat dibuktikan dari banyak bit yang berubah terhadap perubahan pada *plaintext* tidak melebihi dari panjang blok sesuai dengan perilaku dari mode ECB. Pada mode Counter, perubahan hanya mempengaruhi di posisi yang sama dengan bit yang berubah. Sedangkan di mode CBC, perubahan dapat menjalar ke blok berikutnya, sehingga banyak perubahan dipengaruhi oleh di blok berapa bit yang berubah tersebut. Kemudian, banyak bit yang berubah terhadap perubahan bit di *ciphertext* pada *plaintext* bernilai sekitar setengah blok untuk ECB dan CBC dan hanya satu pada mode Counter. Secara umum, terjadi perubahan dengan rasio setengah dari bagian yang dapat dipengaruhi perubahan tersebut.

Dari, analisis terhadap *brute force* dan analisis prinsip *confusion* dan *diffusion* dapat dikatakan bahwa algoritma Brick Cipher yang kami rancang sudah cukup aman.

5. Kesimpulan dan Saran

Hasil simulasi (analisis) menunjukkan bahwa algoritma Brick Cipher yang dirancang oleh penulis memberikan hasil enkripsi dan dekripsi yang baik pada mode ECB, CBC, dan Counter, serta prinsip *confusion* dan *diffusion* dari Shannon pun berhasil terpenuhi pada mode-mode tersebut. Kecepatan enkripsi dari algoritma Brick Cipher ini kurang lebih sama untuk enkripsi dan dekripsi maupun untuk mode ECB, CBC, dan Counter, serta kurang lebih linear terhadap ukuran masukan. Algoritma Brick Cipher juga cukup aman karena membutuhkan waktu yang cukup lama jika dilakukan serangan brute force, serta telah menerapkan prinsip *confusion* dan *diffusion* dari Shannon.

Untuk pengembangan lebih lanjutnya perlu dikembangkan lagi fungsi *scramble*-nya untuk dioptimasi agar algoritma Brick Cipher dapat berjalan lebih cepat.

6. Referensi

[1] R. Munir, *Slide Kuliah IF4020 Kriptografi*, Kriptografi Modern (Bagian 3: Block Cipher), 2020.

[2] C. Shannon, A Mathematical Theory of Cryptography, 1945

[3] Widynski, B. (2017). Middle square Weyl sequence RNG. *arXiv preprint arXiv:1704.00358*.

[4] Chew, Liyana & Ismail, Eddie Shahril. (2020). S-Box Construction Based on Linear Fractional Transformation and Permutation Function. Symmetry. 12. 826. 10.3390/sym12050826.

Acknowledgments

Pertama-tama, penulis mengucapkan puji syukur kepada Allah swt. atas segala nikmat yang telah diberikan-Nya, sehingga penulis bisa menyelesaikan makalah ini tepat pada waktunya. Penulis juga ingin mengucapkan terima kasih kepada dosen mata kuliah IF4020 Kriptografi yang mengajari penulis, khususnya kepada Bapak Dr. Ir. Rinaldi Munir, M.T. yang telah mengajar kelas mata kuliah Kriptografi selama semester ini dan memberikan ilmu kepada penulis. Penulis juga ingin mengucapkan terima kasih kepada keluarga dan teman-teman penulis yang telah memberikan motivasi dan mengirimkan doa kepada penulis.