FADE: *A fusion between Data Encryption Standard and Advanced Encryption Standard*

Willsen Sentosa[1], **Fithratulhay Pribadi**[2].

[1,2] Program Studi Teknik Informatika, Sekolah Teknik Elektro dan Informatika (STEI), Institut Teknologi Bandung (ITB), Jalan Ganesha 10, Bandung 40132
E-mail: 13517036@std.stei.itb.ac.id, 13517140@std.stei.itb.ac.id

Abstraksi. Di dalam jurnal kami, kami membuat sebuah algoritma *block cipher* baru bernama FADE dengan panjang *block* sebesar 256 bit, dan panjang kunci sebesar 128 bit. Pembuatan *block cipher* ini ditujukan sebagai pengganti dari nilai UTS pada mata pelajaran IF 4020 Kriptografi.

Kata kunci : *block cipher*, kriptografi, *encrypt*, *decrypt*

1. Pendahuluan

Sejak zaman dahulu, keamanan pesan merupakan sebuah hal yang penting karena pesan dapat berisi dari sesuatu yang trivial sampai sesuatu yang sangat rahasia. Karena keamanan pesan dianggap sangat penting, manusia memikirkan sebuah cara untuk menyembunyikan pesan. Dari permasalahan tersebut, terciptalah sebuah metode dimana pesan hanya dapat dibaca oleh pengirim dan penerima. Sekarang, bidang yang mempelajari metode ini disebut juga sebagai Kriptografi.

Kriptografi ini kian berkembang seiring waktu, terutama setelah adanya komputer pertama yang diciptakan, kriptografi menjadi sebuah keharusan dalam pengiriman data yang bersifat konfidensial. Namun seiring berkembangnya kriptografi ini, ada orang-orang yang ingin mencuri data/pesan tersebut. Mereka berusaha mencari kelemahan dari metode kriptografi yang telah dibuat. Mereka disebut juga kriptanalis.

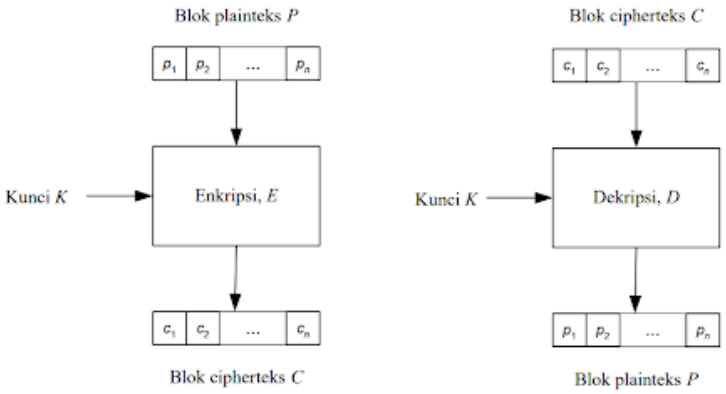
Karena hal ini, maka diperlukan sebuah algoritma kriptografi modern yang kompleks sehingga data-data ini tidak dapat didekripsi oleh kriptanalis dalam jangka waktu yang singkat. Lalu IBM membuat sebuah algoritma kriptografi yang dibuat menjadi sebuah standar. Algoritma ini disebut sebagai *Data Encryption Algorithm*, atau lebih dikenal sebagai *Data Encryption Standard* (DES). Namun seiring berjalannya waktu, algoritma DES ini dianggap kurang aman karena algoritma DES ini dapat diserang menggunakan *brute force attack* oleh komputer-komputer yang terhubung secara nirkabel. Maka perlu dibuat sebuah algoritma yang dapat dijadikan standar yang baru.

Vincent Rijman dan Joan Daemen menjadi tokoh yang membuat sebuah algoritma baru yang saat ini dikenal sebagai *Advanced Encryption Standard* (AES). Sampai saat ini, algoritma AES belum secara resmi berhasil dipecahkan oleh kriptanalis dalam jangka waktu yang pendek. Jika sebuah komputer super cepat yang dapat melakukan *brute force attack* dapat mencoba 1 juta kunci setiap 1 milidetik, maka masih dibutuhkan $5,4\ x\ 10^{18}$ tahun untuk mencoba seluruh kunci.

Algoritma FADE yang kami buat didasari dengan algoritma DES dan AES yang menjadi standar dalam pengenkripsian data. Algoritma ini merupakan gabungan dari DES dan AES yang menerapkan prinsip struktur Feistel, dan prinsip *diffusion and confusion* dari Shannon yang akan mempersulit kriptanalis dalam melakukan serangan kepada algoritma yang kami kembangkan.

2. Studi Pustaka

Algoritma *Block Cipher* adalah algoritma yang membagi teks ke dalam blok-blok bit yang memiliki panjang yang sama.Pada algoritma ini, setiap blok akan dienkripsi dan didekripsi dengan fungsi dan kunci yang sama seperti yang dapat dilihat pada Gambar 1. Enkripsi merupakan proses penyandian pesan asli menjadi pesan yang tidak dapat diartikan seperti aslinya, sedangkan dekripsi merupakan proses merubah pesan yang sudah disandikan menjadi pesan asli (Aribowo & Faqih, 2014). Pesan asli biasa disebut *plaintext* dan pesan hasil penyandian biasa disebut *ciphertext*.



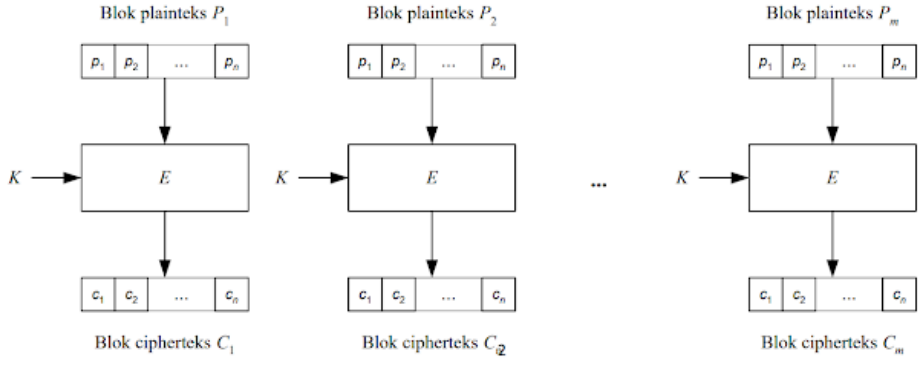
Gambar 1. Skema Enkripsi dan Dekripsi pada *Cipher* Blok (Munir, 2020)

2.1. Mode Operasi *Block Cipher*

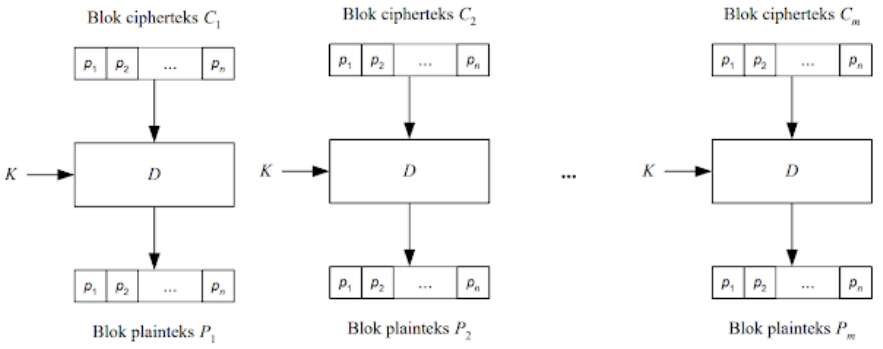
Cara blok-blok *cipher* dioperasikan sebelum/sesudah proses enkripsi/dekripsi disebut mode operasi. Algoritma *Block Cipher* memiliki 5 mode operasi, yaitu *Electronic Code Book, Cipher Block Chaining, Cipher Feedback, Output Feedback,* dan *Counter Mode*. Pada jurnal ini, mode operasi yang akan dibahas adalah mode operasi *Electronic Code Book*, *Cipher Block Chaining*, dan *Counter Mode*.

1. *Electronic Code Book* (ECB)

   Seperti yang dapat dilihat pada Gambar 2 dan 3, pada mode operasi ECB setiap blok *plaintext* dienkripsi secara individual dan independen dari blok lainnya. Mode Operasi ini diberikan nama "*code book"* karena setiap blok *plaintext* yang bernilai sama selalu dienkripsi menjadi blok *ciphertext* yang sama, sehingga secara teoritis dimungkinkan pembuatan buku kode *plaintext* dan *ciphertext* yang saling berkoresponden. Namun, ukuran buku kode yang dibutuhkan akan semakin besar sebanding dengan besar ukuran. Selain itu, kunci yang berbeda juga akan membutuhkan buku yang berbeda.



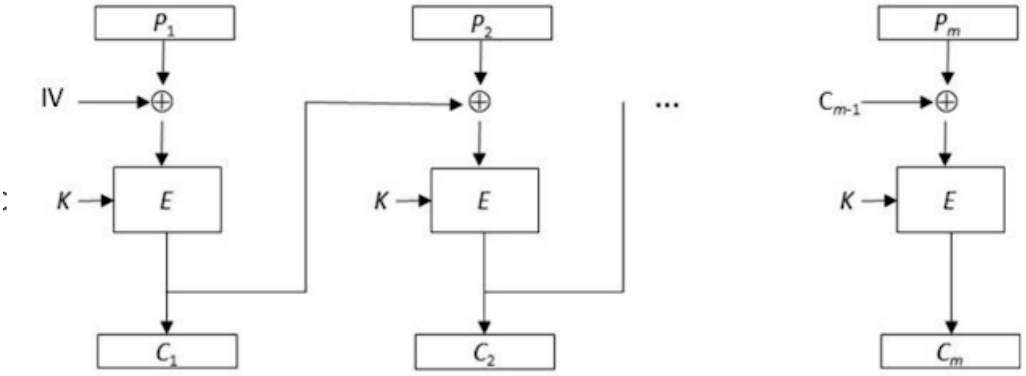
Gambar 2. Skema Enkripsi Mode ECB (Munir, 2020)



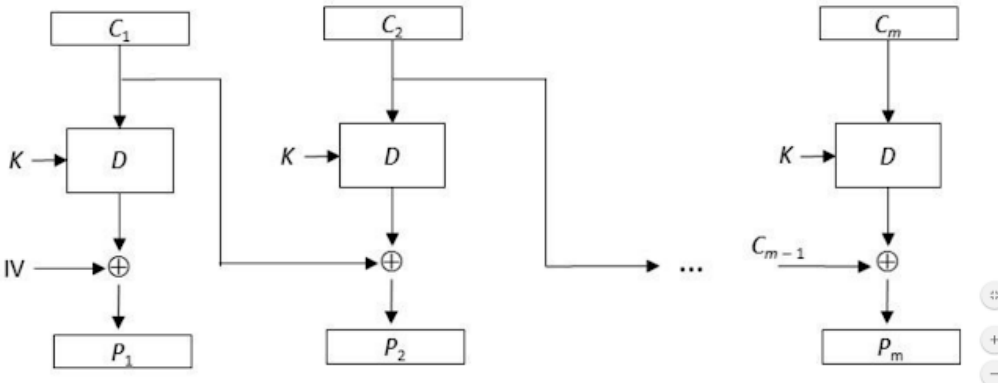
Gambar 3. Skema Dekripsi Mode ECB (Munir, 2020)

2. *Cipher Block Chaining* (CBC)

Berbeda dengan mode ECB, mode CBC membuat sebuah blok *ciphertext* bergantung pada blok-blok *plaintext* sebelumnya. Hal ini dilakukan dengan membuat hasil enkripsi sebuah blok menjadi umpan balik dalam enkripsi blok setelahnya. Sebelum memasuki proses enkripsi, akan dilakukan proses XOR terlebih dahulu antara blok *plaintext* umpan balik tersebut. Setelah itu, barulah hasil dari proses XOR dimasukkan ke dalam fungsi enkripsi. Pada pemrosesan blok pertama, umpan balik yang digunakan merupakan blok semu yang biasa disebut dengan *initialization vector* (IV). Pada saat dekripsi, proses XOR akan dilakukan setelah blok *ciphertext* melewati fungsi dekripsi. Skema proses enkripsi dan dekripsi pada mode CBC dapat dilihat pada Gambar 2 dan 3.



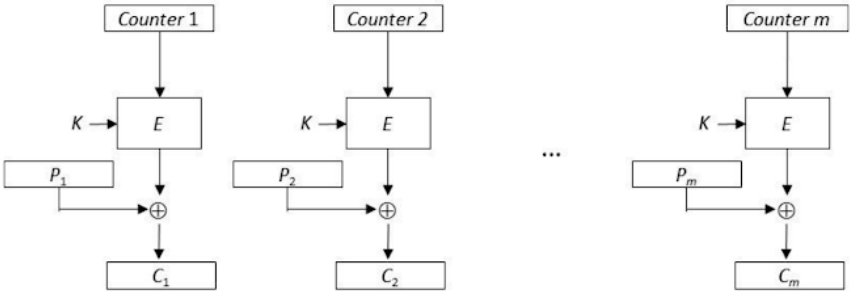
Gambar 4. Skema Enkripsi Mode CBC (Munir, 2020)



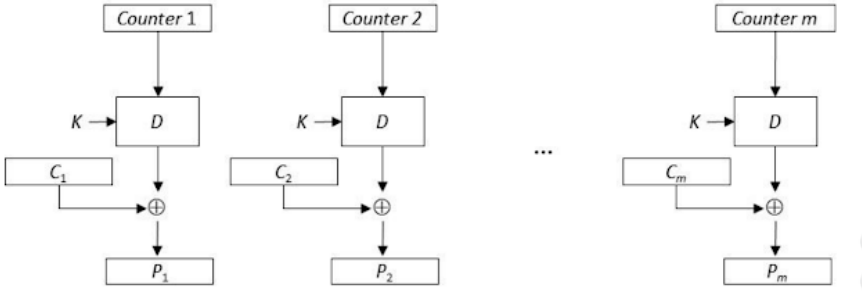
Gambar 5. Skema Dekripsi Mode CBC (Munir, 2020)

3. *Counter*

Pada mode *Counter,* yang dimasukkan ke dalam fungsi enkripsi bukanlah blok *plaintext* melainkan *counter*. *Counter* adalah sebuah nilai berupa blok bit yang memiliki ukuran sama dengan ukuran blok *plaintext*. Pada pemrosesan blok *plaintext* pertama, nilai *counter* akan diinisialisasi terlebih dahulu. Pada pemrosesan blok-blok selanjutnya, nilai *counter* akan dinaikkan sebanyak 1 pada setiap bloknya. Dalam mode operasi ini, blok *ciphertext* adalah hasil XOR antara blok *plaintext* dengan *counter* yang sudah dienkripsi. Untuk mengembalikan *ciphertext* menjadi *plaintext*, yang perlu dilakukan adalah proses XOR antara blok *ciphertext* dengan *counter* yang sudah melewati fungsi dekripsi. Gambaran proses enkripsi dan dekripsi pada mode *Counter* dapat dilihat pada Gambar 6 dan 7.



Gambar 6. Skema Enkripsi Mode *Counter* (Munir, 2020)



Gambar 7. Skema Enkripsi Mode *Counter* (Munir, 2020)

2.2. Prinsip Perancangan *Block Cipher*

Dalam merancang sebuah *block cipher*, terdapat 4 prinsip yang perlu diperhatikan :

1. *Confusion* dan *diffusion*

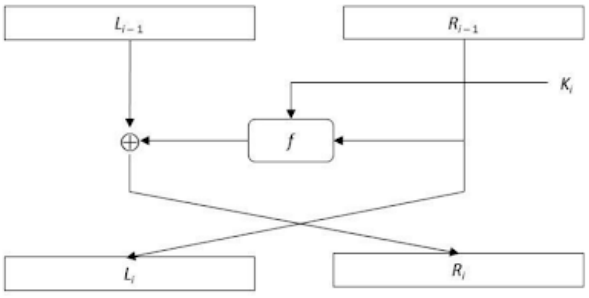
   Prinsip *confusion* dan *diffusion* diperkenalkan oleh Claude Shannon dalam makalahnya pada tahun 1949. Prinsip ini bertujuan untuk membuat kriptanalisis metode statistik menjadi rumit. *Confusion* bermakna penyembunyian seluruh hubungan yang ada di antara *plaintext*, *ciphertext*, dan kunci, sedangkan *diffusion* bermakna penyebaran pengaruh 1 bit *plaintext*/kunci ke sebanyak mungkin *ciphertext*.

2. *Cipher* berulang

   Prinsip *cipher* berulang bermakna pengulangan fungsi transformasi yang mengubah *plaintext* menjadi *ciphertext*. Pada setiap putaran akan digunakan *subkey*/*round key* (kunci putaran) yang dikombinasikan dengan *plaintext*.

3. Jaringan Feistel

   Jaringan Feistel ditemukan oleh Horst Feistel pada tahun 1970. Model jaringan Feistel akan membagi sebuah blok menjadi 2 bagian, yaitu bagian kiri (L) dan bagian kanan (R). Setelah itu, akan dilakukan pemrosesan terhadap kedua bagian tersebut. Bagian kiri hasil pemrosesan merupakan bagian kanan yang lama, sedangkan bagian kanan hasil pemrosesan merupakan hasil XOR antara bagian kiri yang lama dengan bagian kanan lama yang sudah melalui fungsi transformasi. Skema jaringan Feistel dapat dilihat pada Gambar 8.



Gambar 8. Skema Jaringan Feistel (Munir, 2020)

Jaringan Feistel bersifat *reversible* dimana fungsi transformasi untuk melakukan enkripsi dan dekripsi merupakan fungsi yang sama. Karena itu, dengan model jaringan Feistel, fungsi transformasi dapat dibuat serumit mungkin.

4. *S-box* (Kotak-S)

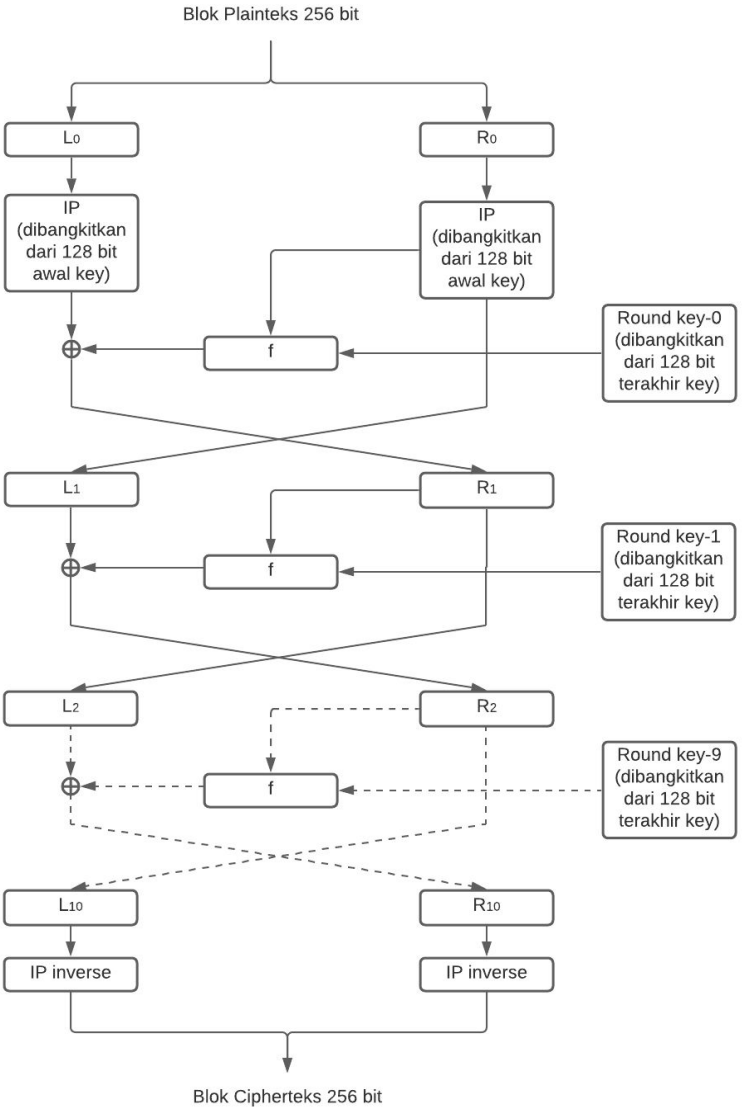
   S-box merupakan matriks yang berisi substitusi sederhana yang memetakan satu atau lebih bit dengan satu atau lebih bit yang lain. *S-box* merupakan satu-satunya langkah *non linear* karena operasinya berupa *look-up table*. Masukan dari operasi ini dijadikan sebagai indeks pada *S-box*, sedangkan luarannya berupa *entry* sesuai indeks pada *S-box*.

3. Algoritma FADE

Algoritma *block cipher* baru yang penulis rancang bernama FADE. Algoritma ini membagi *plaintext* menjadi blok-blok berukuran 256 bit. Kunci yang dibutuhkan oleh algoritma ini juga berukuran 256 bit, dengan 128 bit pertama akan dipakai untuk membangkitkan matriks permutasi, sedangkan 128 bit selanjutnya dipakai untuk membangkitkan *round key*.

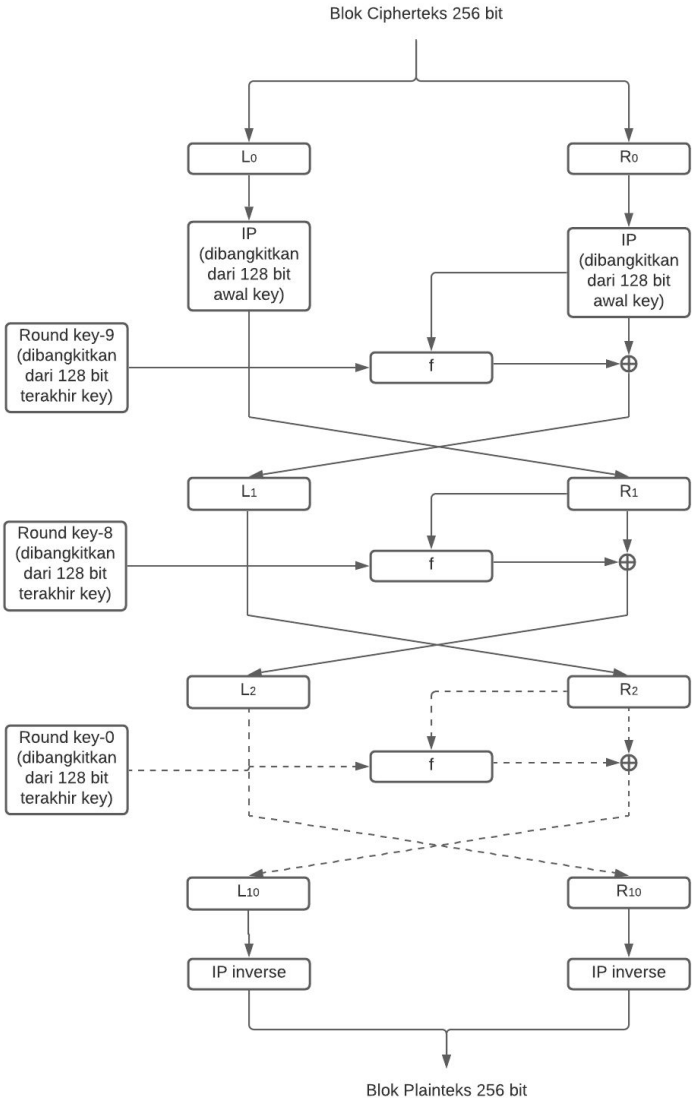
Dalam fungsi enkripsi, setiap blok dari *plaintext* akan kembali dibagi menjadi 2 bagian, yaitu bagian kiri dan bagian kanan. Bagian kiri merupakan 128 bit pertama dan bagian kanan merupakan 128 bit terakhir. Bagian-bagian tersebut lalu akan melalui proses permutasi dengan menggunakan matriks permutasi yang sudah dibangkitkan oleh 128 bit pertama dari kunci. Bagian kiri dan kanan yang sudah melalui proses permutasi lalu masuk ke dalam jaringan Feistel yang berulang sebanyak 10 kali. Setelah keluar dari jaringan Feistel, dilakukan proses permutasi *inverse* pada masing-masing bagian tersebut. Blok *ciphertext* merupakan gabungan dari bagian kiri dan kanan yang sudah melalui permutasi *inverse*.

Jaringan Feistel yang ada pada algoritma ini berbentuk seperti jaringan Feistel pada umumnya. Bagian kiri dan kanan yang menjadi masukan akan diproses pada jaringan tersebut. Bagian kiri luaran merupakan bagian kanan pada masukan, sedangkan bagian kanan luaran merupakan hasil XOR bagian kiri masukan dengan bagian kanan masukan yang sudah melalui fungsi transformasi *f* dengan *round key* sesuai indeks pengulangan.



Gambar 9. Skema Enkripsi Algoritma FADE

Karena memakai jaringan Feistel, proses dekripsi *ciphertext* mirip dengan proses enkripsi *plaintext*. Perbedaan antara proses enkripsi dan dekripsi hanya terdapat pada urutan melakukannya saja. Pada proses dekripsi, *ciphertext* dibagi menjadi blok-blok berukuran 256 bit. Setiap blok *ciphertext* dibagi menjadi bagian kiri dan bagian kanan. Setelah itu, dilakukan proses permutasi untuk seluruh bagian. Hasil dari proses permutasi dimasukkan ke jaringan feistel terbalik dan diulangi sebanyak 10 kali. *Round key* yang dipakai pada dekripsi ini juga dibalik urutannya. Sebagai contoh, pada iterasi dengan indeks 0 menggunakan *round key* dengan indeks 9. Bagian kanan dari luaran merupakan bagian kiri masukan. Bagian kiri dari luaran merupakan hasil XOR bagian kanan masukan dengan bagian kiri masukan yang sudah melalui fungsi transformasi *f* dengan *round key* yang sesuai. Setelah melewati jaringan feistel terbalik, kembali dilakukan permutasi *inverse* pada bagian kiri dan bagian kanan. Hasil dari permutasi *inverse* lalu digabungkan sehingga membentuk blok *ciphertext* seperti semula.



Gambar 10. Skema Dekripsi Algoritma FADE

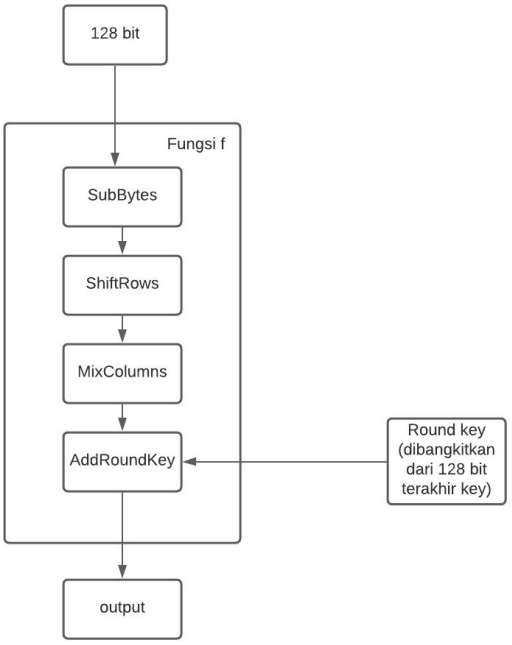
Bagian yang melewati fungsi transformasi *f* dibentuk menjadi matriks *hexadecimal* dengan baris dan kolom berukuran 4. Matriks tersebut kemudian melalui 4 tahap yang terdapat pada fungsi transformasi *f*:

1. *Sub Bytes*
   Tahap *sub bytes* merupakan tahap untuk melakukan operasi substitusi dengan memetakan setiap *byte* dalam representasi *hexadecimal* dari masukan menggunakan *S-box*. *Byte* dengan representasi XY akan disubstitusi dengan *entry* pada *S-box* pada baris ke-X dan kolom ke-Y. *S-box* yang digunakan sama dengan *S-box* pada Algoritma Rijndael.

2. *Shift Rows*
   Tahap *Shift Rows* merupakan tahap dimana baris-baris larik *state* akan digeser secara *wrapping*, dimana baris *state* kedua akan digeser satu kali ke kiri, baris *state* ketiga akan digeser dua kali, dan baris *state* keempat akan digeser tiga kali

3. *Mix Columns*
   Tahap *Mix Columns* merupakan tahap dimana data akan diacak pada masing-masing kolom larik *state*. *Mix Column* yang dilakukan menyerupai yang dilakukan pada algoritma AES, yaitu tiap kolom larik *state* akan diubah menjadi hasil kali larik *Rcon* dengan kolom larik *state*.

4. *Add Round Key*
   Tahap *Add Round Key* adalah tahap dimana larik dari *state* akan dilakukan operasi XOR dengan *round key*.



Gambar 11. Fungsi Transformasi *f* Algoritma FADE

Pada Algoritma FADE, terdapat beberapa mode yang dapat digunakan, yaitu:

1. *Electronic Code Book* (ECB)
   Pada mode operasi ECB, setiap blok dienkripsi secara individual dan independen dari blok lainnya, sehingga tidak terdapat perlakukan khusus pada blok yang akan dienkripsi ataupun didekripsi. *Plaintext* yang sudah diubah ke dalam blok-blok berukuran 256 bit akan langsung masuk ke dalam proses enkripsi, begitu pula blok hasil proses enkripsi langsung digabung sehingga membentuk *ciphertext*.

2. *Cipher Block Chaining* (CBC)

   Pada mode operasi CBC, blok *plaintext* yang akan memasuki proses enkripsi akan di-XOR terlebih dahulu dengan hasil enkripsi pada blok sebelumnya. Pada blok pertama, proses XOR tersebut akan dilakukan blok *plaintext* dengan IV (*initialization vector*) dari masukan pengguna. Agar dapat melakukan operasi XOR tersebut, IV harus berukuran sama dengan blok *plaintext,* yaitu berukuran 256 bit. Saat mengembalikan *ciphertext* menjadi *plaintext*, hasil proses dekripsi suatu blok *ciphertext* di-XOR terlebih dahulu dengan blok *ciphertext* sebelumnya. Setelah di-XOR, barulah blok-blok tersebut dapat digabung membentuk *plaintext*. Sama seperti saat akan melakukan enkripsi, pada blok pertama, hasil dekripsi di-XOR dengan IV dari pengguna.

3. *Counter*

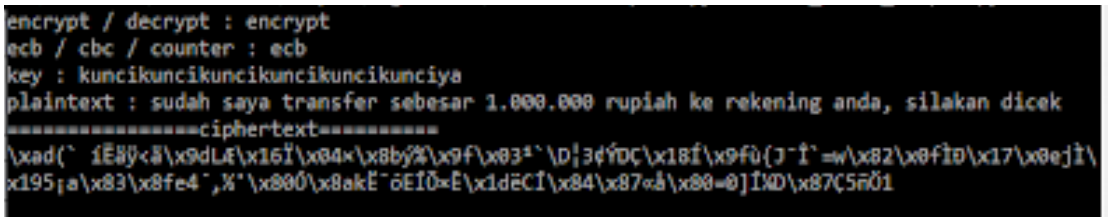
   Pada mode *counter*, digunakan sebuah *Initialization Vector* untuk memperkuat mode *counter*. *Initialization Vector* ini diambil dari key, tepatnya 128 bit akhir dari key. Kemudian *Initialization Vector* ini akan ditambahkan 128 bit counter yang dimulai dari 0 sehingga menghasilkan 256 bit. Pada tiap block ciphertext yang ada, counter akan ditambahkan nilainya sebesar 1 tiap pengulangannya. Pada tiap pengulangan pengenkripsian block cipher, counter akan dienkripsi menggunakan algoritma FADE, lalu hasilnya akan dilakukan operasi XOR dengan bit plaintext sehingga menghasilkan sebuah block cipher text.

4. Eksperimen dan Analisis Hasil

4.1. Contoh tampilan program

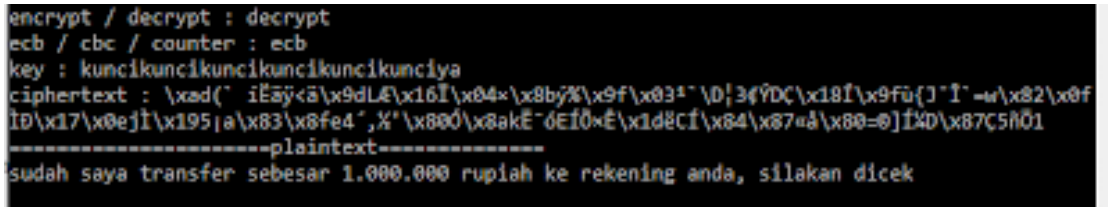
Berikut beberapa contoh tampilan program :

1. Enkripsi ECB



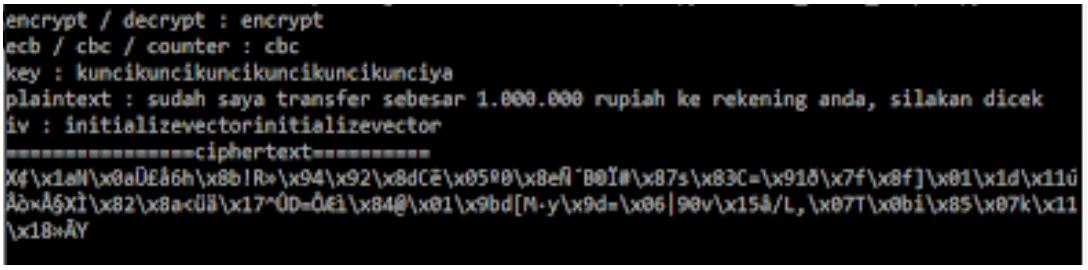
Gambar 12. Tampilan Enkripsi ECB

2. Dekripsi ECB



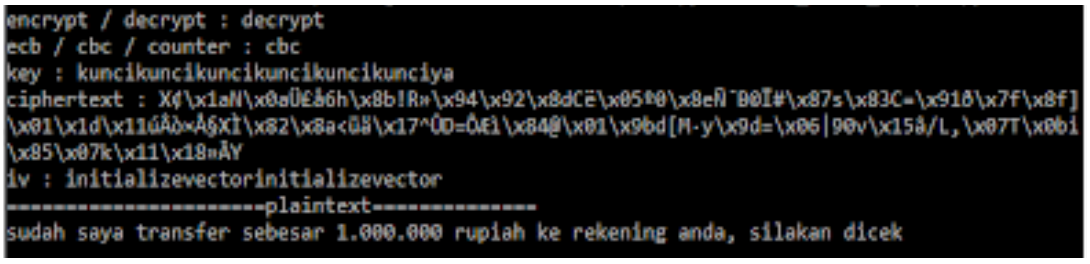
Gambar 13. Tampilan Dekripsi ECB

3. Enkripsi CBC



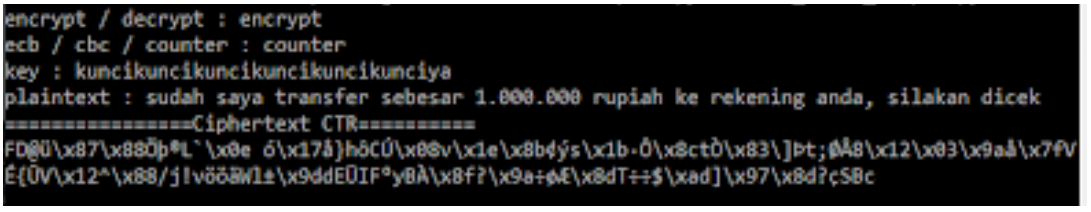
Gambar 14. Tampilan Enkripsi CBC

4. Dekripsi CBC



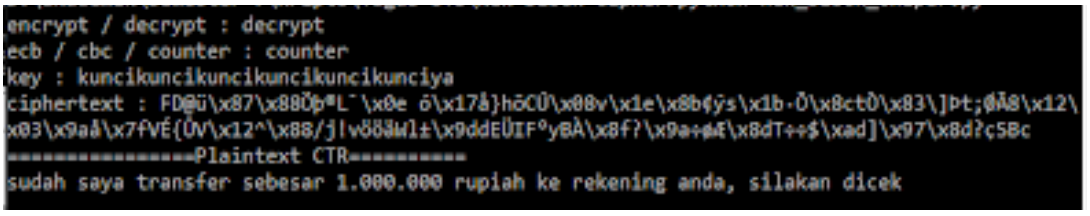
Gambar 15. Tampilan Dekripsi CBC

5. Enkripsi *Counter*



Gambar 16. Tampilan Enkripsi *Counter*

6. Dekripsi *Counter*



Gambar 17. Tampilan Enkripsi *Counter*

4.2. Eksperimen

Dalam menguji algoritma dan program yang telah dibuat, penulis melakukan 3 eksperimen :

1. Eksperimen yang dilakukan menggunakan plaintext yang berbeda tiap ukurannya, dan kunci sebesar 256 bit.

Plaintext : Ini adalah pesan rahasia Bob untuk Alice

Key : Jin0XZeO3JjaE9XDW4M9EOspgGXhFsjK

Hasil Cipher Text(ECB):
\x91\x86fi\x1aA.õÜnÍÖ@\x8c\x99\x8fmS&àþuMe.ÍV\x1d\x82&\x95¾\x8925·\x00\x88)Öù³\x12¿\x84(v\x8cÑ\x04©±é~õD¬_d\x7f{j4\x01

IV (CBC) : sAgH0MiidQrlXW5cWBg7SwQzF3BPbHkJ
Hasil Cipher Text(CBC):
½K>z21^Jy\x82ª\x86\x89&,rÎo"\x99\x1fÒb3Í\x17ßMë+Pê;Î\x07H±\x9e(\x84Ü8òó±½=è`¦'>\x1941Ö\x88Òö^âä\x1b

Hasil Cipher Text(Counter):
ú3¿ã¤\x84\x12-\x94\x84\x7f5î\x9aÍ÷:S¡1b\x88\x96\x87:¦\x0bUèO\>\x0cj\x854qêÈAAÝ\x0dè<«\x1dwd³@t^Å\x17r\x0e\x84\x7fèðì§®

Plaintext: Sejak jaman dulu kala, keamanan pesan merupakan sebuah hal yang penting karena pesan dapat berisi dari sesuatu yang *trivial* sampai sesuatu yang sangat rahasia. Karena keamanan pesan dianggap sangat penting, maka manusia memikirkan sebuah cara untuk menyembunyikan pesan. Dari permasalahan tersebut, terciptalah sebuah metode dimana pesan hanya dapat dibaca oleh pengirim dan penerima. Sekarang, bidang yang mempelajari metode ini disebut juga sebagai Kriptografi.

Key: IndonesiaRayaMerdekaMerdekaHidup
Hasil Cipher Text (ECB):
Ris\x0eßD\x98µÙoÞúú¨\x10f\x87\x97@!Ú\x9e\x1fÏeñ²aR®\x1fÏüëÈMêð\x16\x956\x90\x9fµ\x15S¡\x0d\x0fU\x01\x0c\x9aU\x83?*uRûË[ñ(ãÀÔ×½á,¢u´]â÷\x11\x90Þ\x10¼\x9bYñ\x9aý\x8c³B\x04\x02[_\x9e\x1fD~\x87\x87\x9a¸0Á^)\x93O\x96\x07ÜFCÝp÷óß\x80½Eð#oÐ/ÈD_i\x14Ü'©E\x9f\x8f\x14Ï\x8d\x1d1\x0b\x0c.¹Cceíjq|+\x86q\x0c9&¿\x9b\x83áþ©º\x8eÊ{d\xad¥$¾'L\x09\x93\x85c÷\x8a\x8bäx~ÒM\x0b\x16Q\x0bÈÙ\x85®J=\xa0ª\x12SIì|N\x9cI÷p\x17\x02§îÆ6\x8aI\x8bG¥*\x85³N\x1aà¶kßþ)õ?7ì¼P\x1d\x139°uT©\x14\x9dÅ\x9c'3¿!cÏj2\x00K=hy\x96\x1eH-ëGdµÓ1ðÄ?;Û\x81"NÂ¯O&\x02x"\x18¿»½@\x05FÜcæS\x84+\x14e\x00£×]§\x0b\x89ëÎ6±Z\x0d\x02²\xad\x9fó\x91õ\x9faJ\x86Bî\x17Ç´0jÖî\x9f\x17À\x08ßÏÜç9\x80®T*ð¹ä\x99©«¸í\x0ff;Ëâ>@¦PM\x1a\x9cs»yµï\x9aå>ç\x92»\x07ãw\x81À\x09º\x896\x9b\x86\x04×\x82É\x10÷\x86\x8c#ó^ªÙ÷\x15¹\x8eM\x1fO(¾í(\x0bXô&"úú_j\x00\x82ø©Ç¬U\My´\x94!ùçÔ\x9e~ç\x09\x0cá¥³\x09OHo\x98\x9dtY\x90H\xad¯¹°lAã`(Ëzy}3¡º\x17iÔ/\x80\x16

IV (CBC): IndonesiaTanahAirkuTanahTumpahDa
Hasil Cipher Text (CBC):
\x7f\x02\x99W\x9cî2°bß}©\x19ØË\x0e:\x80\x80wg½y\x9d8i¯ÔDmú\x07R>K\x9b\x98\x9d0\x1f{\x97Ï6`\x9f=ü\x92Ä\x87\x9f÷Ó-»\x10ú.\x1fT-6vø\xadD¢NTt\x0a+\x9bÐ}TK\x0bw\x0bë'\x83h\x14®\x84Ä¼\xadË\x9a1/£HñM7[\x001\x83wc¸&àu\x1eæèÀiÝÐ¿-m¾þâ¸\x1e3~\x99oá\x0cXÍªya4+\x01Í\x92Ïa%¥\x84fGVåyæ#\xa0Ü#jï©î:c\x1e²:\x16<ªá\x05\x89\xadÈ\x1f\x8fE£»:_ÝóÔ³DþÌ\x9d\x90sk6\x16M÷·\x9bKí\x19UäT\x9e\x1d¡az4\x84Û\x85Þ\x87#÷ø&ÏÙ{8õ\x8aÕùá-A¾¼\x194VéÐgíÜ/!¶A)ý\x84\x14q\x99Ú\x7fÇ\x00÷\x1f\x09ÙÍå/äÍTûò\x9cà~´\x9fä§u\x0b\x91N=où\x14¶YáUÏ¤\x92\x93â¹n))._c\x11\x81ámÞ.gûp\x91\x1f%Ù\x8c\x15°\x1c+\x9c+$XoÛ\x122Ä\x1evXî>nÆºÓYñ\x85¼¨²?6\x90qaí&A\x1d\x08üU)-Î\x9d4j0¥\x80v·\x08W¬}SË´\x1b\x037\x1a\x84\x0dñÁ¡g×Í¯Ü¦AÓVÇ\x92\x98\x9déM\x0c6Ù^o9fä(ß\x98\x13\x0bï§\x8bê½\x9cï7¹\x9a\x8d=\x9eY\xadþOéä.6y¡\x04ºåøÁ\x92jÈËõ´¹ª¨\x81@\x96§½\x9e\x80hp~ß\x9b¬\x04NÂCÈ\x12\x19åÂ\x15"\x7f+Yº8í5B\x85\x92ªt\\0f

Hasil Cipher Text(Counter):
\x13©ÐÁx\x12+.\x00×\x0aÇ\x08`\x1aÆò+§\x96z@\x16\x8e\x92%SyÃ\x8aì¦\x94æ\x11v2ò
t¿\x8d¯ÖîS~÷ÈùUù=-ôÀJ5\xad=\x0eQýä\E\x09\x92\x80Ã%£Ð\x7f\x165ä\x1føÀ4$ß\x12ø
w\x82\x1e\xa0b\x1e2~®.-\x99¼\x9bÇ{ìA\x86µ'\x18Ç-cì(þ\x1ePû8
Ì\x19òð'\x95r\x03áOÝ£»»*Á\x84¡Ì\x84^ú@¡bÑ'\x94dwo²\x95³é\x05Ó\x0eáã\x18\x1bÂ\ß÷
>¸\x02\\\x9c¸Îæ\x99x^;'¦YÌÄÖÞ\x1eÉwY-\x10\x9býÏ-I\xad\x8fL¾}¦Äï\x1f¹ÿ\x95CtºÛ6¬£M
²0z\x0dóÑ&FkJ\x11\x92\x9cµ
\x10)a\x17|\\x9fk*1¡¶Ç3Ò\x9e\x09ì;\x19)øCé¥î"\x90OÜ\x8cÌNÈ!êªC3aB\x0c\x80K(Ns!³\x
88ãÈ\x19°\x8bì\x1bÓHÚ15\x83uî%\x8d\x81\x10\xa0U\x1b16\x88ÚÕ×\x1eX³#\x0b5(\x9aª
û\x82õQ£½]r^§Ðy\x92\x8a\x03\x0bG\x1b®Ô\x83Â\x8d\x00~:æ\x83_J\x05\x9dÔ*\x0a\x9
b\x8b\x89\x94µ~Ã\x89~ÑĬj<Ð\x08-\x1d\x0f!\x11\x8dò\x04\x94³s\xad«úE\xad¬ö\x15\x8cg
r?Í¦·3þd\x0f\x9bÕ\x0céªTHá\x03\x0e\x084uä\x91rKu¿bô\x01Y3\x02q«\x01\x97\x13\x99\x
12\x86ô
6#©uÏâr(\x0f\x82{Äïh`\x8d|z\x18\x07tóÉn6î\x87\x9e\x16Z\x99/;\x8dÜ}Î\x7f.Y`ær\x99\x8
bÖÜ§ª

2. Eksperimen yang dilakukan dengan mengubah 1 bit kunci saat dekripsi

Ciphertext :
- **ECB :**
  \x91\x86fi\x1aA.õÜnÍÖ@\x8c\x99\x8fmS&àþuMe.ÍV\x1d\x82&\x95¾\x8925·\x00
  \x88)Öù³\x12¿\x84(v\x8cÑ\x04©±é~õD¬_d\x7f{j4\x01
- **CBC:**
  ½K>z21^Jy\x82ª\x86\x89&,rÎo"\x99\x1fÒb3Í\x17ßMë+Pê;Î\x07H±\x9e(\x84Ü8òó
  ± ½=è`¦'>\x1941Ö\x88Òö^âä\x1b
- **Counter:**
  ú3¿ã¤\x84\x12-\x94\x84\x7f5î\x9aÍ÷:S¡1b\x88\x96\x87:¦\x0bUèO\>\x0cj\x854qêÈ
  AAÝ\x0dè<«\x1dwd³@t^Å\x17r\x0e\x84\x7fèðì§®

Key Asli: Jin0XZeO3JjaE9XDW4M9EOspgGXhFsjK

Hasil Plaintext (ECB):
Ini adalah pesan rahasia Bob untuk Alice

IV (CBC) : sAgH0MiidQrlXW5cWBg7SwQzF3BPbHkJ
Hasil Plaintext (CBC):
Ini adalah pesan rahasia Bob untuk Alice

Hasil Plaintext (Counter):
Ini adalah pesan rahasia Bob untuk Alice

Ciphertext :
- **ECB :**
  \x91\x86fi\x1aA.õÜnÍÖ@\x8c\x99\x8fmS&àþuMe.ÍV\x1d\x82&\x95¾\x8925·\x00
  \x88)Öù³\x12¿\x84(v\x8cÑ\x04©±é~õD¬_d\x7f{j4\x01
- **CBC:**
  ½K>z21^Jy\x82ª\x86\x89&,rÎo"\x99\x1fÒb3Í\x17ßMë+Pê;Î\x07H±\x9e(\x84Ü8òó
  ± ½=è`¦'>\x1941Ö\x88Òö^âä\x1b
- **Counter:**
  ú3¿ã¤\x84\x12-\x94\x84\x7f5î\x9aÍ÷:S¡1b\x88\x96\x87:¦\x0bUèO\>\x0cj\x854qêÈ
  AAÝ\x0dè<«\x1dwd³@t^Å\x17r\x0e\x84\x7fèðì§®

Key Modifikasi : Êin0XZeO3JjaE9XDW4M9EOspgGXhFsjK

Hasil Plaintext (ECB):
×eñúÞDRäÐÉK\x0bOæxèwK$\x15s\x1a\x9b\x88\x85
º+éa\x8c\xady\x1f\x97i3Ù\x8d\x80É´\x8c9\x90ãå'ÉA\x0eË\x89®1u\x92Â2Îk[~$

IV (CBC) : sAgH0MiidQrlXW5cWBg7SwQzF3BPbHkJ
Hasil Plaintext (CBC):
³\x1cÌ\x94G\x16Ã¡lzU\x92\x9f¶x\x1f:ª\x954\x00f_Asã&Ð¨§|\x19IV,½è\x7f\x1cVø§\x85Ìý
+w¦xé\x1bì}¡\x8fÍ\x8a\x0c\x08&\x8c3å\x16

Hasil Plaintext (Counter):
³\x83Ã\x86\x93|h\x85\x87õ2j·\x9fz\x9b×RÕ\x06pÝ\x8b\x8a\x86Æ\x85\xad#%\x8eÞ¼VõV
\x81«è¾>ËÊ\x84\x943Q!ÎÁn0ÕîT3è\x87nÊ\x18{y\x0b

3. Eksperimen dengan modifikasi sedikit *plaintext*

Plaintext: Transfer aku uang sebanyak 58.000 dollar amerika
Plaintext modifikasi : Transfer aku uang sebanyak 50.000 dollar amerika
Key: Tn4tvVNSKoB11ICc8rBwHjQefntx0CC3

Hasil CipherText(ECB):
ýöTÝ3|$ñ±ÿ-@ëº\xad\x87'\x8báÊ\x0d7ä^àÛFÑ\x13&\x1b¥ÖN\x15~.y\x00Bg\x82\x1f9ê\
Wg\x1dÿ-3\x87£w&\x07\x7fd\x8a\x84ù\x98|
CipherText Hasil Modifikasi Plaintext (ECB):
JÛ\x06ü³òèÈ\x03M:¿Q\x92:å\x08\x82ûÆL¦Z&~ÈSä?éC~ÖN\x15~.y\x00Bg\x82\x1f9ê\Wg
\x1dÿ-3\x87£w&\x07\x7fd\x8a\x84ù\x98|

IV(ECB):
ULr2nC367Nm2E7yyoQfu4SEoz8XNkgEM

Hasil CipherText(CBC):
×\x19äz|îku¸'3\x8a\x0eêZ\x1d\x16|1ÂËteØiie\x9fï\x87ñw$Ã+°¤yKG\x09±\x05qÉÝ\x9awÁ
\x9e¹,"¥$§6Fdp¯j\x97]
CipherText Hasil Modifikasi Plaintext (CBC):
Þ(¬y\x94¸I\x0aø\x89\x86VÄ\x9aï\x1aÎï\x06Æð9\x8dxO$}»\Êò\xa0µ#×Iò\x8b¼\x00Z½4_
A\x13\x06\x15©\x8ej\x8e£È\x8b\x14»b\x84\x9e\x10ÍQ9

Hasil CipherText(Counter):
\x89ÿ'r\x1be\xadÉåöØ\x04¸â=X\x06§8\xa0\x18ý¤íÌ]eæÅ+ZÇ-w\4fp\x14ï®ù%
Ïí\x84{ÜØS5×ø\x1c£\x11Þ\x94ÉH\x94_)
CipherText Hasil Modifikasi Plaintext (Counter):
\x89ÿ'r\x1be\xadÉåöØ\x04¸â=X\x06§8\xa0\x18ý¤íÌ]eæÍ+ZÇ-w\4fp\x14ï®ù%
Ïí\x84{ÜØS5×ø\x1c£\x11Þ\x94ÉH\x94_)

4. Eksperimen dengan modifikasi *ciphertext*

Plaintext: Transfer aku uang sebanyak 58.000 dollar amerika
Key: Tn4tvVNSKoB11ICc8rBwHjQefntx0CC3

Hasil CipherText(ECB):
ýöTÝ3|$ñ±ÿ-@ëº\xad\x87'\x8báÊ\x0d7ä^àÛFÑ\x13&\x1b¥ÖN\x15~.y\x00Bg\x82\x1f9ê\
Wg\x1dÿ-3\x87£w&\x07\x7fd\x8a\x84ù\x98|
Modifikasi CipherText(ECB):
ýöTÝ4|$ñ±ÿ-@ëº\xad\x87'\x8báÊ\x0d7ä^àÛFÑ\x13&\x1b¥ÖN\x15~.y\x00Bg\x82\x1f9ê\
Wg\x1dÿ-3\x87£w&\x07\x7fd\x8a\x84ù\x98|
Hasil Dekripsi(ECB):
f\xad\x80éSn&MQÛ\x84\x0a\x13k;2òNº°Ç\\x0d^1ÄÙ@\{EÔ0 dollar amerika

IV(ECB):
ULr2nC367Nm2E7yyoQfu4SEoz8XNkgEM

Hasil CipherText(CBC):
×\x19äz|îku¸'3\x8a\x0eêZ\x1d\x16|1ÂËteØiie\x9fï\x87ñw$Ã+°¤yKG\x09±\x05qÉÝ\x9awÁ
\x9e¹,"¥$§6Fdp¯j\x97]
Modifikasi CipherText(CBC):
×\x19äz|îka¸'3\x8a\x0eêZ\x1d\x16|1ÂËteØiie\x9fï\x87ñw$Ã+°¤yKG\x09±\x05qÉÝ\x9awÁ
\x9e¹,"¥$§6Fdp¯j\x97]
Hasil Dekripsi(CBC):
\x9c\x17b\x80Ò\x9auÿ\x0e'÷:\x10P\x88¥\x11µ\x87"#'Í(\x93ý^\x0e8æÉG0 dollaf
amerika\x00\x00\x00q

Hasil CipherText(Counter):
\x89ÿ'r\x1be\xadÉåöØ\x04¸â=X\x06§8\xa0\x18ý¤íÌ]eæÅ+ZÇ-w\4fp\x14ï®ù%
Ïí\x84{ÜØS5×ø\x1c£\x11Þ\x94ÉH\x94_)
Modifikasi CipherText(Counter):
\x89ÿ'r\x1br\xadÉåöØ\x04¸â=X\x06§8\xa0\x18ý¤íÌ]eæÅ+ZÇ-w\4fp\x14ï®ù%
Ïí\x84{ÜØS5×ø\x1c£\x11Þ\x94ÉH\x94_)
Hasil Dekripsi(Counter):
Transqer aku uang sebanyak 58.000 dollar amerika

4.3. Analisis Hasil

Pada eksperimen pertama, program melakukan enkripsi *plaintext* dengan ukuran 320 bit menjadi *ciphertext* dengan ukuran 512 bit serta enkripsi *plaintext* dengan ukuran 3704 bit menjadi *ciphertext* dengan ukuran 3840 bit. Dari kedua pengujian tersebut, dapat diketahui bahwa pada semua mode operasi, ukuran *ciphertext* yang dihasilkan merupakan kelipatan 256 pertama setelah ukuran suatu *plaintext*, atau dapat ditulis sebagai berikut :

$$ukuran\_ciphertext = ((ukuran\_plaintext \; div \; 256) + 1) * 256$$

Pada eksperimen kedua, dilakukan uji coba pengubahan kunci sebanyak 1 bit pada saat melakukan dekripsi. Kunci yang asli adalah “Jin0XZeO3JjaE9XDW4M9EOspgGXhFsjK” dan kunci yang telah dimodifikasi sebanyak 1 bit adalah “Êin0XZeO3JjaE9XDW4M9EOspgGXhFsjK”. Dari hasil yang didapat, terlihat bahwa pada semua mode operasi, hanya dengan mengubah 1 bit pada kunci, *plaintext* hasil dekripsi yang didapat berbeda jauh dengan *plaintext* asli.

Pada eksperimen ketiga, pengujian dilakukan dengan mengubah 1 *byte plaintext* saat akan melakukan enkripsi. Dari hasil yang didapat, terlihat bahwa pada mode operasi ECB, pengubahan 1 *byte* akan mengubah 1 blok *ciphertext.* Pada mode operasi CBC, pengubahan 1 *byte* akan berpengaruh pada blok-blok setelahnya. Sedangkan pada mode operasi *counter*, pengubahan 1 *byte* hanya akan

berpengaruh pada 1 *byte ciphertext*. Dari hasil-hasil tersebut, dapat disimpulkan bahwa mode operasi yang paling menerapkan prinsip *diffusion* merupakan mode ECB.

Pada eksperimen keempat, dilakukan pengujian dengan pengubahan 1 *byte ciphertext* pada saat dekripsi. Pada hasil eksperimen dapat dilihat bahwa pada mode ECB, pengubahan *byte* akan berpengaruh pada blok tempat *byte* tersebut berada. Pada mode CBC, pengubahan 1 *byte* akan berpengaruh pada blok tersebut dan blok setelahnya, sedangkan pada mode *counter*, hanya beberapa *plaintext* yang terkena pengaruh perubahan. Hal ini disebabkan karena hanya *counter* yang diproses oleh algoritma FADE, sedangkan plaintext hanya di XOR kan dengan *counter*. Maka perubahan cipher text tidak akan mempengaruhi secara signifikan terhadap plain text dalam mode *counter*.

Ukuran kunci yang dipakai untuk melakukan enkripsi dan dekripsi pada algoritma FADE adalah 256 bit. Untuk melakukan *bruteforce attack*, terdapat 2^256 kemungkinan kunci yang harus dicek. Jika berasumsi ada yang ingin melakukan *bruteforce attack* dengan kemampuan komputasi melakukan percobaan sebanyak 10 juta kunci per detik, maka diperlukan waktu sekitar 3.7 x 10^62 tahun untuk mencoba semua kemungkinan kunci. Karena itu, dapat disimpulkan bahwa algoritma ini aman dari *bruteforce attack*. Selain itu, pada algoritma ini juga dilakukan cukup banyak substitusi dan pergeseran yang akan menambah kesulitan untuk kriptanalis yang ingin menyerang pesan yang dienkripsi.

5. Kesimpulan dan Saran Pengembangan

5.1. Kesimpulan

Berikut adalah kesimpulan yang penulis dapatkan setalah membuat dan menguji algoritma serta program *block cipher* baru bernama FADE :

1. Ukuran *ciphertext* yang dihasilkan merupakan kelipatan 256 pertama setelah ukuran suatu *plaintext* yang menjadi masukan.
2. Pada seluruh mode operasi yang tersedia, pengubahan 1 bit kunci saat dekripsi akan berpengaruh pada keseluruhan *plaintext* yang dihasilkan.
3. Mode operasi yang paling menerapkan prinsip *diffusion* adalah mode CBC karena pengubahan 1 *byte plaintext* berpengaruh pada blok *ciphertext* tempatnya berada serta blok-blok setelahnya.
4. Pada mode operasi ECB pengubahan 1 *byte ciphertext* saat dekripsi akan berpengaruh pada keseluruhan blok tempat *byte* tersebut berada. Pada mode operasi CBC, pengubahan tersebut akan berpengaruh juga pada blok *plaintext* setelahnya. Sedangkan pada mode operasi *counter*, pengubahan 1 *byte* hanya akan berpengaruh pada 1 *byte plaintext*.
5. Algoritma FADE aman dari *bruteforce attack*.

5.2. Saran Pengembangan

Untuk pengembangan selanjutnya, algoritma FADE ini dapat dibuat lebih fleksibel, terutama dalam panjang kunci yang saat ini hanya dapat dilakukan menggunakan 256 bit. Selain itu, dapat dikembangkan juga mode operasi lain untuk diterapkan pada algoritma ini.

6. References

[1] Aribowo E dan Faqih M R 2014 *Visualisasi Algoritma Chiper Block Chaining Sebagai Media Pembelajaran Berbasis Mobile Android* (Yogyakarta : Universitas Ahmad Dahlan)

[2] Adin B P dan Aulia F 2016 *Blox: Algoritma Block Cipher* (Bandung: Institut Teknologi Bandung)

[3] Munir R 2020 *Bahan Kuliah IF4020 Kriptografi* (Bandung: Institut Teknologi Bandung)

[4] Joan Daemen and Vincent Rijmen, The Design of Rijndael, AES - The Advanced Encryption Standard, Springer-Verlag 2002 (238 pp.)

Acknowledgments

Penulis ingin berterima kasih kepada Tuhan yang Maha Esa atas berkat dan anugerahNya sehingga penulis dapat memiliki kesempatan untuk membuat sebuah algoritma *block cipher* baru. Penulis juga ingin berterima kasih kepada Institut Teknologi Bandung yang telah memfasilitasi kegiatan belajar mengajar kami. Kami berterima kasih kepada dosen khususnya kepada bapak Rinaldi Munir selaku dosen pengajar IF 4020 Kriptografi yang telah mengajari kami mengenai Kriptografi. Dan yang terakhir, penulis berterima kasih kepada keluarga dan teman-teman kami yang mendukung kami dalam pembuatan makalah ini.