Vanilla Block Cipher

Letivany Aldina

Program Studi Teknik Informatika, Sekolah Teknik Elektro & Informatika, Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132 Indonesia
E-mail: 13514067@std.stei.itb.ac.id

Abstract. Algoritma Vanilla Block Cipher merupakan algoritma *block cipher* yang menggunakan DES *(Data Encryption Standard)* sebagai basis dalam konsep dan implementasi. Algoritma Vanilla Block Cipher beroperasi dalam blok berukuran 64 bit dengan panjang minimal kunci juga 64 bit atau 8 *bytes*. Algoritma Vanilla Block Cipher menerapkan prinsip *confusion* dan *diffusion* agar kriptanalisis sulit dilakukan. Dalam makalah ini juga dilakukan pembahasan simulasi dan hasil serta analisis keamanan pada algoritma Vanilla Block Cipher.

Keywords. *kriptografi, vanilla block cipher, confusion, diffusion, jaringan feistel, kotak-s, enkripsi, dekripsi.*

1. Pendahuluan

1.1. Latar Belakang

Kriptografi adalah ilmu yang mempelajari teknik-teknik matematika yang berhubungan dengan aspek keamanan informasi seperti kerahasiaan, integritas data, otentikasi entitas, dan otentikasi asal usul data (Menezes, 1996). Kriptografi sangat diperlukan terutama dalam keamanan pengiriman dan penerimaan data dan informasi dari suatu pihak ke pihak lain.

Kriptografi sudah ada sejak zaman dahulu kala, dan terus mengalami perkembangan sampai saat ini. Pada awal kemunculan kriptografi, metode-metode yang digunakan masih bersifat sederhana. Semakin berkembangnya kebutuhan manusia terhadap keamanan informasi, maka metode-metode dalam kriptografi juga semakin berkembang. Jenis kriptografi yang banyak diterapkan pada saat ini disebut dengan kriptografi modern. Sedangkan metode-metode kriptografi periode sebelumnya disebut dengan kriptografi klasik.

Pada makalah ini, penulis akan membahas sebuah algoritma kriptografi modern yakni *block cipher*. Algoritma *block cipher* ini disebut dengan Vanilla Block Cipher karena terinspirasi dari salah satu jenis algoritma block cipher, DES, yang menerapkan prinsip-prinsip *block cipher*. Algoritma ini memanfaatkan struktur Feistel yang dilakukan dalam 16 kali putaran, prinsip *confusion* dengan menerapkan kotak-S, dan prinsip *diffusion* dengan menerapkan permutasi, baik pada pesan yang akan dienkripsi maupun pada kata kunci yang akan digunakan.

1.2. Review Beberapa Algoritma Block Cipher Lainnya

Pada bagian ini akan dibahas secara singkat algoritma-algoritma *block cipher* lainnya.

1.2.1. DES

DES, atau *Data Encryption Standard*, yang menjadi inspirasi penulis dalam menulis makalah ini, adalah standar atau aturan enkripsi yang dikembangkan oleh IBM pada tahun 1972 berdasarkan algoritma Lucifer yang dikembangkan oleh Horst Feistel. DES beroperasi pada blok berukuran 64 bit dan enkripsi dilakukan dalam 16 kali putaran. Setiap putaran menggunakan kunci-kunci internal yang berbeda-beda, dimana kunci-kunci internal tersebut dibangkitkan dari kunci eksternal yang diperoleh dari pengguna. Secara umum, setiap blok diproses dengan cara melakukan permutasi awal, kemudian dilakukan 16 kali putaran enkripsi, dan inversi permutasi awal.

1.2.2. GOST

GOST atau *Gosudarstvenny Standard*, merupakan algoritma enkripsi yang dikembangkan oleh Uni Soviet pada tahun 1970 yang memiliki struktur yang mirip dengan DES. Pada algoritma ini terdapat 32 putaran dan kunci internal yang digunakan sebanyak 8 buah. Sehingga perbedaan prinsip kerja GOST dengan DES terletak pada penggunaan kunci internal ini, yakni pada GOST 8 buah kunci internal tersebut dijadwalkan penggunaannya.

1.2.3. 3DES

3DES merupakan salah satu variasi dari DES, dimana pada algoritma ini enkripsi dilakukan sebanyak tiga kali. Pada 3DES ini kita dapat menggunakan dua atau tiga buah kunci eksternal.

1.2.4. AES

AES atau *Advanced Encryption Standard* merupakan standar algoritma kriptografi yang berbasis kunci publik. AES dikembangkan berdasarkan algoritma kriptografi yang dikembangkan oleh Rijndael pada 2001. Perbedaan utama antara AES dengan DES adalah pada AES algoritma kriptografi beroperasi dalam orientasi *byte*.

1.3. Pendekatan Vanilla Block Cipher

Vanilla Block Cipher menggunakan pendekatan DES (*Data Encryption Standard*) sebagai basis dalam konsep dan implementasi. Vanilla Block Cipher beroperasi dalam blok berukuran 64 bit. Iterasi jaringan Feistel dilakukan dalam 16 kali putaran. Prinsip *confusion* dari Shannon diterapkan menggunakan kotak-S dan prinsip *diffusion* diterapkan menggunakan sejumlah operasi permutasi.

2. Dasar Teori

2.1. Block Cipher

Block cipher merupakan salah satu jenis algoritma kriptografi modern dimana pesan dibagi-bagi ke dalam sejumlah blok dengan panjang yang sama. Kunci yang digunakan dalam enkripsi memiliki panjang yang sama dengan ukuran blok.

2.1.1. Mode Block Cipher

2.1.1.1. Electronic Code Book (ECB)

Pada mode ini setiap blok-blok pesan atau plainteks dienkripsi secara individual dan independen menjadi blok-blok cipherteks. Blok plainteks yang sama selalu dienkripsi menjadi blok cipherteks yang sama. Keuntungan dari mode ECB ini adalah blok-blok plainteks dienkripsi secara individual dan independen sehingga kita tidak perlu mengenkripsi file secara linear. Kekurangan dari mode ECB ini adalah mudah diserang secara statistik karena blok-blok plainteks yang sama menghasilkan blok-blok cipherteks yang sama pula.

2.1.1.2. Cipher Block Chaining (CBC)

Mode CBC merupakan mode block cipher yang membuat ketergantungan antar blok, dimana setiap blok cipherteks bergantung pada blok plainteksnya dan seluruh blok plainteks sebelumnya. Kelebihan dari mode CBC ini adalah blok plainteks yang sama menghasilkan blok cipherteks yang tidak sama, sehingga kriptanalisis terhadap mode CBC ini akan lebih sulit dibandingkan mode ECB. Kekurangan mode CBC ini adalah kesalahan enkripsi pada sebuah blok plainteks akan menyebabkan kesalahan beruntun pada blok cipherteksnya dan seluruh blok cipherteks berikutnya.

2.1.1.3. Cipher Feedback (CFB)

Mode CFB merupakan mode yang mengenkripsi data dalam unit yang lebih kecil dari ukuran blok. Sehingga cara kerja proses enkripsinya sama dengan *stream cipher*. Kekurangan mode CFB ini adalah kesalahan enkripsi pada sebuah blok plainteks akan berpengaruh pada blok cipherteksnya dan seluruh blok cipherteks berikutnya.

2.1.1.4. Output Feedback (OFB)

Mode OFB merupakan pengembangan dari mode CFB dimana n-bit dari hasil enkripsi terhadap antrian disalin menjadi elemen posisi paling kanan pada antrian. Hal tersebut menyebabkan kesalahan enkripsi dan dekripsi pada sebuah blok plainteks hanya akan berpengaruh pada blok cipherteks yang berkoresponden saja.

2.1.1.5. Counter Mode

Mode *counter* merupakan mode yang tidak menggunakan prinsip perantaian. Pada mode ini ditetapkan sebuah variabel *counter* yang berisi blok bit berukuran sama dengan ukuran blok plainteks. Nilai variabel *counter* akan diinisialisasi pada blok pertama dan nilainya diinkremen satu untuk blok-blok berikutnya.

2.1.2. Prinsip Block Cipher

2.1.2.1. Prinsip Confusion dan Diffusion Shannon

Confusion merupakan prinsip menyembunyikan hubungan antara kunci dan cipherteks. Prinsip *confusion* ini menyebabkan proses kriptanalisis sulit dilakukan karena kunci sulit ditemukan dari cipherteks. Perubahan satu bit dalam kunci akan berpengaruh pada seluruh cipherteks. Dalam penerapannya, prinsip *confusion* diimplementasikan menggunakan operasi substitusi.

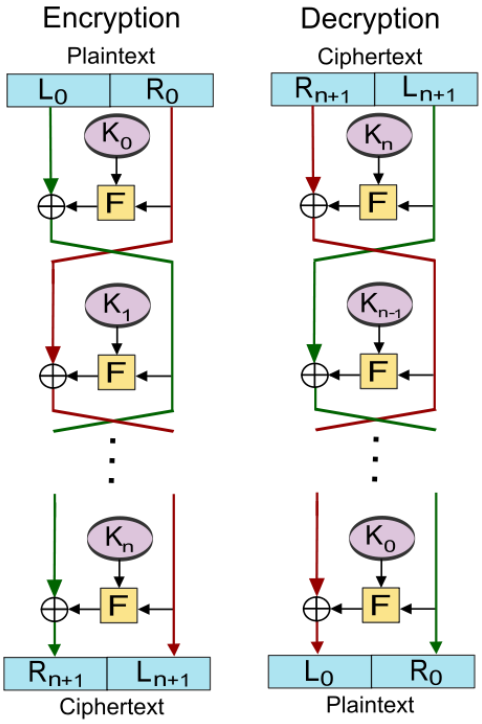
Diffusion merupakan prinsip menyembunyikan hubungan antara plainteks dan cipherteks. Perubahan satu bit pada plainteks akan berpengaruh pada perubahan cipherteks. Dalam penerapannya, prinsip *diffusion* diimplementasikan menggunakan operasi transposisi.

2.1.2.2. Cipher Berulang (Iterated Cipher)

Cipher berulang merupakan prinsip transformasi plainteks yang dilakukan dalam sejumlah putaran. Untuk setiap putaran, digunakan kunci putaran (*round key*) yang dibangkitkan dari kunci eksternal yang diperoleh dari pengguna.

2.1.2.3. Jaringan Feistel

Jaringan Feistel merupakan sebuah struktur simetris *block cipher* dalam melakukan enkripsi dan dekripsi. Prinsip jaringan Feistel ini memungkinkan kita untuk tidak perlu membuat algoritma baru untuk dekripsi. Jaringan Feistel sendiri merupakan sebuah cipher berulang dimana fungsi internalnya disebut dengan fungsi putaran.



Gambar 1. Enkripsi dan Dekripsi pada Jaringan Feistel
Sumber : https://en.wikipedia.org/wiki/Feistel_cipher

2.1.2.4. Kotak-S (S-Box)

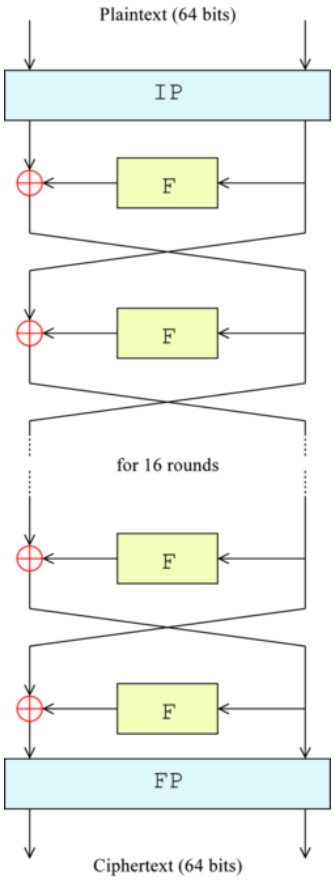
Kotak-S atau Kotak-Substitusi dalam *block cipher* merupakan matriks substitusi yang digunakan untuk mengaburkan hubungan antara kunci dan cipher teks. Sehingga kotak-S ini menerapkan prinsip *confusion* dari Shannon. Dengan menggunakan kotak-S ini, sejumlah *m* bit masukan akan ditransformasi menjadi *n* bit keluaran. Kotak-S diimplementasikan sebagai sebuah tabel pencarian (*lookup table*).

2.2. DES

Pada bagian review algoritma *block cipher* lainnya, telah dijelaskan secara umum mengenai DES. DES beroperasi pada blok berukuran 64 bit. Setiap blok dienkripsi sebanyak 16 kali putaran, dimana setiap putaran menggunakan kunci internal yang berbeda-beda. Kunci-kunci internal dibangkitkan dari kunci eksternal yang diperoleh dari pengguna. Setiap blok akan diterapkan permutasi awal, 16 kali putaran enkripsi serta inversi permutasi awal.

Pada awalnya dilakukan pembangkitan kunci-kunci internal yang dilakukan dengan menggunakan sejumlah operasi permutasi dan operasi pergeseran bit. Tahap permutasi awal bertujuan untuk mengacak urutan bit plainteks sehingga menerapkan prinsip *diffusion* Shannon. Dalam sebuah tahap putaran jaringan Feistel, juga terdapat sejumlah operasi yang dilakukan. Pertama, dilakukan pemisahan blok menjadi sub-blok kanan dan kiri dimana masing-masing sub-blok berukuran 32 bit. Untuk sub-blok kanan, dilakukan ekspansi sub-blok dari ukuran 32 bit menjadi 48 bit. Kemudian dilakukan operasi XOR terhadap hasil ekspansi dengan kunci internal. Jika proses yang akan dilakukan adalah proses enkripsi, maka kunci internal yang digunakan dimulai dari kunci pertama. Sedangkan jika proses yang dilakukan adalah proses dekripsi, maka kunci internal yang digunakan dimulai dari kunci terakhir.

Selanjutnya dilakukan proses substitusi menggunakan kotak-S dengan hasil proses sebelumnya. Hasil dari proses substitusi ini kemudian digunakan untuk proses permutasi. Setelah sub-blok kanan diproses sampai dengan proses permutasi tersebut, operasi XOR dilakukan terhadap hasil proses sub-blok kanan dengan sub-blok kiri. Sub-blok kanan kemudian digabungkan kembali dengan sub-blok kiri. Proses permutasi terakhir atau inversi permutasi awal kemudian dilakukan terhadap hasil penggabungan tersebut.



Gambar 2. Jaringan Feistel pada DES.
Sumber : https://en.wikipedia.org/wiki/Data_Encryption_Standard

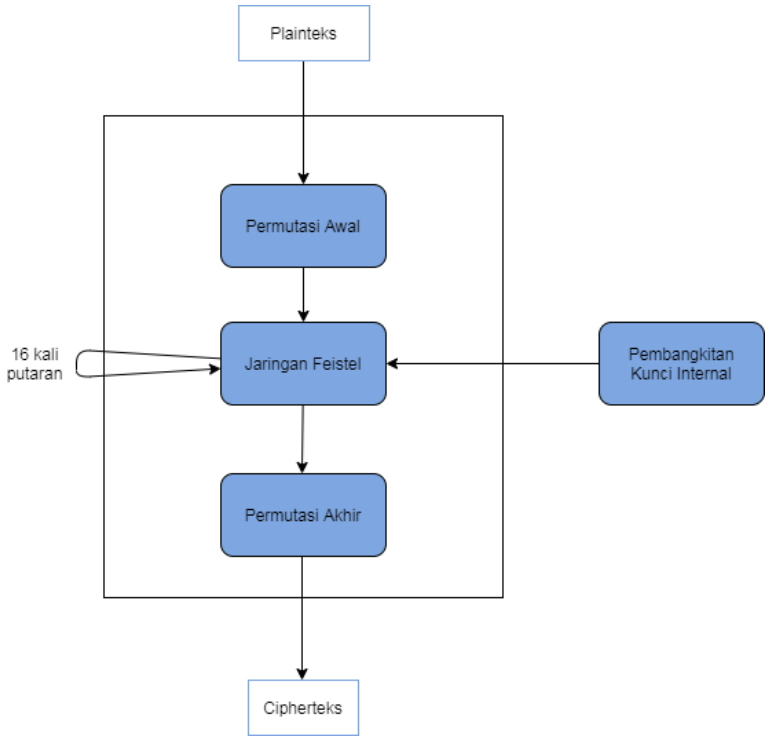
3. Algoritma Vanilla Block Cipher

3.1. Gambaran Umum

Secara umum, algoritma Vanilla Block Cipher pada awalnya akan menerima masukan berupa kunci dan plain teks. Baik kunci maupun plain teks akan dicek panjangnya jika diperlukan penambahan bit *padding* atau tidak. Setelah itu, kunci-kunci internal akan dibangkitkan dari kunci yang diterima sebagai masukan.

Plain teks kemudian akan dibagi menjadi blok-blok yang berukuran 64 bit. Untuk setiap blok, akan dilakukan tahapan proses sebagai berikut.

1. Mengonversi string blok plainteks menjadi bit
2. Melakukan permutasi awal terhadap blok
3. Memisahkan blok menjadi sub-blok kanan dan sub-blok kiri
4. Melakukan 16 kali putaran, dimana untuk setiap putaran akan dilakukan proses sebagai berikut.
    a. Mengekspansi sub-blok kanan yang awalnya berukuran 32 bit menjadi 48 bit.
    b. Melakukan operasi XOR antara hasil ekspansi sub-blok kanan dengan kunci internal. Jika aksi yang akan dilakukan adalah enkripsi, maka kunci internal dimulai dari kunci pertama. Sedangkan jika aksi yang dilakukan adalah dekripsi, kunci internal dimulai dari kunci terakhir.
    c. Melakukan proses substitusi hasil langkah 2 dengan matriks kotak-S.
    d. Melakukan permutasi dari hasil proses substitusi kotak-S pada langkah 3.
    e. Melakukan operasi XOR kembali antara hasil permutasi dengan sub-blok kiri.
    f. Menggabungkan kembali hasil sub-blok kanan dan hasil sub-blok kiri.
    g. Melakukan permutasi akhir, yakni inversi permutasi awal terhadap hasil penggabungan.



Gambar 3. Diagram alur proses umum Algoritma Vanilla Block Cipher

3.2. Pembangkitan Kunci Internal

Terdapat dua langkah utama dalam tahap pembangkitan kunci internal, yakni sebagai berikut.

1. Tahap awal

   Pada tahap awal ini, dilakukan permutasi awal terhadap kunci eksternal. Hasil permutasi tersebut kemudian dibagi menjadi kunci sub-blok kanan dan sub-blok kiri

2. Tahap iterasi

   Pada tahap iterasi sebanyak 16 kali putaran, dilakukan pergeseran bit untuk masing-masing sub-blok kanan dan sub-blok kiri. Hasilnya kemudian digabungkan kembali dan dilakukan permutasi untuk mendapatkan kunci pada putaran tersebut.

3.3. Proses Substitusi

Proses substitusi dalam algoritma Vanilla Block Cipher dilakukan untuk menerapkan prinsip *confusion* dari Shannon. Proses substitusi ini dilakukan dengan menggunakan kotak-S. Terdapat 8 matriks kotak-S yang masing-masing berukuran 4 x 16. Elemen-elemen setiap matriks kotak-S memiliki nilai yang sama dengan elemen-elemen matriks kotak-S yang digunakan pada DES.

3.4. Proses Permutasi

Terdapat 3 langkah permutasi yang dilakukan dalam algoritma Vanilla Block Cipher ini, yakni sebagai berikut.

1. Permutasi awal

   Permutasi awal dilakukan terhadap bit-bit plainteks menggunakan matriks permutasi awal. Tujuannnya adalah untuk mengacak urutan dari bit-bit plainteks tersebut.

2. Permutasi setelah substitusi

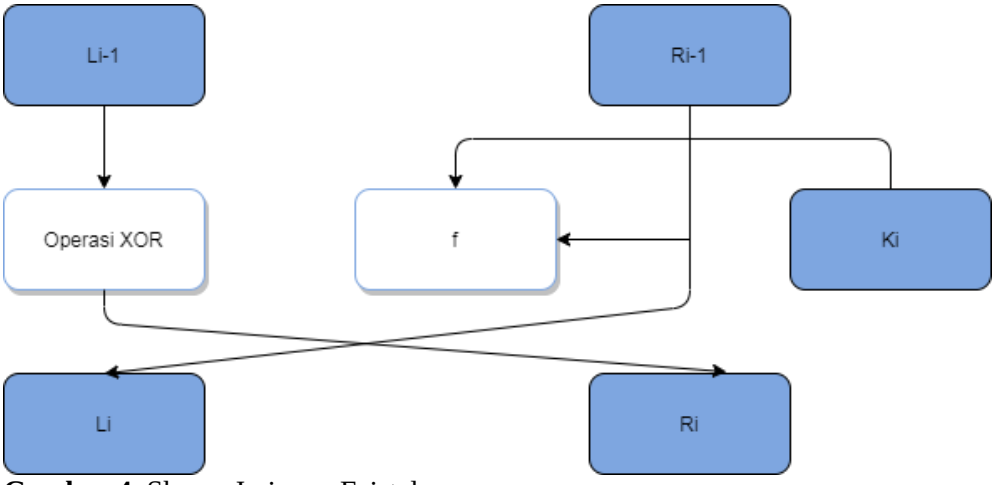
   Langkah permutasi ini dilakukan setelah proses substitusi dengan kotak-S dilakukan. Tujuannya adalah untuk mengacak urutan dari bit-bit hasil proses substitusi.

3. Permutasi akhir

   Permutasi akhir atau inversi permutasi dilakukan sebagai langkah terakhir untuk membuat algoritma Vanilla Block Cipher ini semakin sulit. Permutasi akhir ini dilakukan setelah hasil proses sub-blok kanan digabungkan dengan hasil proses sub-blok kiri dalam jaringan Feistel.

3.5. Jaringan Feistel

Dalam algoritma Vanilla Block Cipher ini, jaringan Feistel diimplementasikan sebagai cipher berulang sebanyak 16 kali putaran untuk setiap blok. Perbedaan aksi antara enkripsi dan dekripsi diimplementasikan pada proses ini. Jika aksi yang akan dilakukan adalah enkripsi, maka kunci internal diproses dimulai dari kunci pertama. Jika aksi yang akan dilakukan adalah dekripsi, maka kunci internal diproses dimulai dari kunci terakhir. Langkah-langkah dalam jaringan Feistel dapat diilustrasikan melalui Gambar 4 berikut.



Gambar 4. Skema Jaringan Feistel

4. Simulasi dan Pembahasan Hasil

Dalam pengujian yang dilakukan terhadap algoritma Vanilla Block Cipher dilakukan tiga simulasi dengan menggunakan kunci dan plain teks yang dapat dilihat pada Tabel 1 berikut.

Tabel 1. Kunci dan plain teks

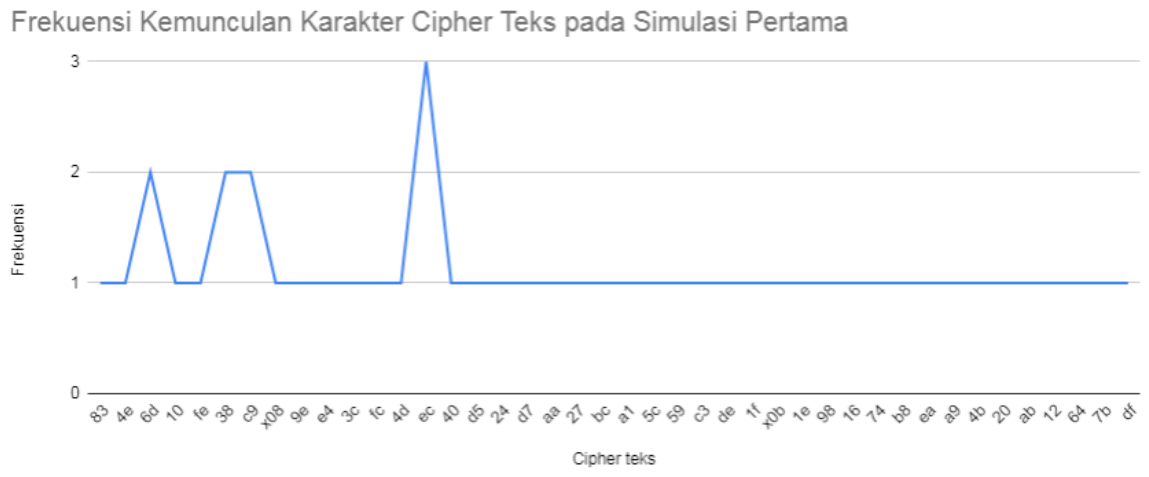
Kunci	`ada kunci algoritma vanilla block cipher`
Plain teks	`ada prinsip confusion dan diffusion dari shannon`

4.1. Simulasi Pertama

Simulasi pertama dilakukan secara normal tanpa ada perubahan bit. Representasi kunci, plain teks, dan cipher teks dapat dilihat pada Tabel 2. Statistik frekuensi kemunculan karakter dapat dilihat dalam grafik pada Gambar 5.

Tabel 2. Kunci, plain teks, dan cipher teks dalam heksadesimal

Kunci	`61 64 61 20 6b 75 6e 63 69 20 61 6c 67 6f 72 69 74 6d 61 20 76 61 6e 69 6c 6c 61 20 62 6c 6f 63 6b 20 63 69 70 68 65 72`
Plain teks	`61 64 61 20 70 72 69 6e 73 69 70 20 63 6f 6e 66 75 73 69 6f 6e 20 64 61 6e 20 64 69 66 66 75 73 69 6f 6e 20 64 61 72 69 20 73 68 61 6e 6e 6f 6e`
Cipher teks	`83 4e 6d 10 fe 38 c9 e2 08 9e e4 c9 3c fc 4d ec 40 6d d5 24 d7 ec aa ec 38 27 bc a1 5c 59 c3 de 1f 0b 1e 98 16 74 b8 ea a9 4b 20 ab 12 64 7b df`



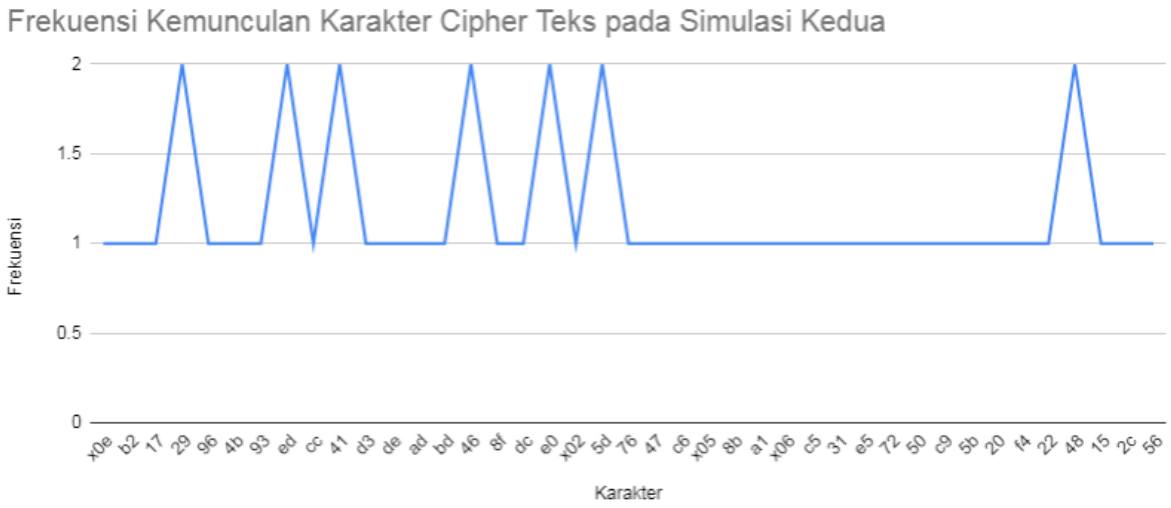
Gambar 5. Grafik frekuensi kemunculan karakter dalam cipher teks pada simulasi pertama

4.2. Simulasi Kedua

Simulasi kedua dilakukan dengan melakukan perubahan 1 bit pada kunci. Pada Tabel 3 dapat dilihat perubahan secara keseluruhan terhadap cipher teks dibandingkan cipher teks pada simulasi pertama. Statistik frekuensi kemunculan karakter dapat dilihat dalam grafik pada Gambar 6.

Tabel 3. Hasil perubahan 1 bit pada kunci

Kunci (awal)	ada kunci algoritma vanilla block cipher
	61 64 61 20 6b 75 6e 63 69 20 61 6c 67 6f 72 69 74 6d 61 20 76 61 6e 69 6c 6c 61 20 62 6c 6f 63 6b 20 63 69 70 68 65 72
Kunci	bda kunci algoritma vanilla block cipher
	62 64 61 20 6b 75 6e 63 69 20 61 6c 67 6f 72 69 74 6d 61 20 76 61 6e 69 6c 6c 61 20 62 6c 6f 63 6b 20 63 69 70 68 65 72
Cipher teks (awal)	83 4e 6d 10 fe 38 c9 e2 08 9e e4 c9 3c fc 4d ec 40 6d d5 24 d7 ec aa ec 38 27 bc a1 5c 59 c3 de 1f 0b 1e 98 16 74 b8 ea a9 4b 20 ab 12 64 7b df
Cipher teks	0e b2 17 29 96 4b 93 ed cc 41 d3 de ad bd 46 8f 46 dc e0 02 41 29 5d 76 47 e0 c6 05 8b a1 06 c5 31 e5 72 ed 50 c9 5b 20 f4 22 5d 48 15 48 2c 56



Gambar 6. Grafik frekuensi kemunculan karakter dalam cipher teks pada simulasi kedua

4.3. Simulasi Ketiga

Simulasi ketiga dilakukan dengan melakukan perubahan 1 bit pada plain teks. Pada Tabel 4 dapat dilihat sebagian perubahan pada cipher teks dibandingkan dengan cipher teks pada simulasi pertama. Frekuensi kemunculan karakter cipher teks pada simulasi ketiga dapat dilihat pada Gambar 7.

Tabel 4. Hasil perubahan 1 bit pada plain teks

Plain teks (awal)	ada prinsip confusion dan diffusion dari shannon
	61 64 61 20 70 72 69 6e 73 69 70 20 63 6f 6e 66 75 73 69 6f 6e 20 64 61 6e 20 64 69 66 66 75 73 69 6f 6e 20 64 61 72 69 20 73 68 61 6e 6e 6f 6e
Plain teks	bda prinsip coffusion dan diffusion dari shannon
	62 64 61 20 70 72 69 6e 73 69 70 20 63 6f 6e 66 75 73 69 6f 6e 20 64 61 6e 20 64 69 66 66 75 73 69 6f 6e 20 64 61 72 69 20 73 68 61 6e 6e 6f 6e
Cipher teks	83 4e 6d 10 fe 38 c9 e2 08 9e e4 c9 3c fc 4d ec 40 6d d5 24 d7 ec aa ec 38 27 bc a1 5c 59 c3 de 1f 0b 1e 98 16 74 b8 ea a9 4b 20 ab 12 64 7b df
	28 14 e8 30 d5 91 4c 2c 08 9e e4 c9 3c fc 4d ec 40 6d d5 24 d7 ec aa ec 38 27 bc a1 5c 59 c3 de 1f 0b 1e 98 16 74 b8 ea a9 4b 20 ab 12 64 7b df

Gambar 7. Grafik frekuensi kemunculan karakter dalam cipher teks pada simulasi ketiga

5. Analisis Keamanan

5.1. Analisis Confusion dan Diffusion

Dari simulasi kedua dan ketiga yang dilakukan pada bagian Simulasi dan Pembahasan Hasil dapat dilihat bahwa perubahan 1 bit pada kunci dan plain teks menyebabkan perubahan yang cukup signifikan pada cipher teks. Perubahan yang cukup signifikan tersebut akan mempersulit proses yang dilakukan dalam kriptanalisis karena prinsip *confusion* dan *diffusion* diterapkan dalam algoritma Vanilla Block Cipher ini.

5.2. Analisis Serangan Brute Force

Algoritma Vanilla Block Cipher menggunakan kunci dengan panjang minimal 64 bit. Sehingga terdapat minimal 2^64 kunci yang harus dicoba untuk dapat menemukan kunci yang digunakan.

5.3. Analisis Statistik

Dari ketiga simulasi yang telah dilakukan frekuensi kemunculan karakter dalam cipher teks pada setiap simulasi rata-rata kemunculan karakter adalah satu. Hanya terdapat sedikit karakter yang muncul dua atau tiga kali. Sehingga proses kriptanalisis terhadap algoritma Vanilla Block Cipher secara statistik akan cukup sulit dilakukan karena kemunculan karakter dalam cipher teks hampir terdistribusi rata.

6. Kesimpulan dan Pengembangan

Algoritma Vanilla Block Cipher menggunakan DES (*Data Encryption Standard)* sebagai basis dalam konsep dan implementasi. Dari simulasi dan analisis yang telah dilakukan Algoritma Vanilla Block Cipher memiliki prinsip *confusion* dan *diffusion* dalam penerapannya sehingga kriptanalisis akan cukup sulit dilakukan.

Algoritma Vanilla Block Cipher dapat dikembangkan ke depannya dengan menerapkan operasi-operasi tambahan. Operasi-operasi dalam algoritma Vanilla Block Cipher sebagian besar berupa operasi permutasi, substitusi dan XOR dalam mengimplementasikan prinsip *confusion* dan *diffusion*.

7. References

[1] A. Menezes, P. van Oorshot, dan S. Vanstone . 1996. *Handbook of Applied Cryptography.* CRC Press.

[2] Munir, Rinaldi. 2020. Slide Kuliah IF4020 Kriptografi : Kriptografi Modern.

[3] Munir, Rinaldi. 2020. Slide Kuliah IF4020 Kriptografi : Algoritma Kriptografi Modern.

[4] https://crypto.stackexchange.com/questions/5492/brute-force-attack-on-des-property-of-des diakses pada tanggal 11 Maret 2020.

[5] https://crypto.interactive-maths.com/frequency-analysis-breaking-the-code.html diakses pada tanggal 11 Maret 2020.

Pernyataan

Penulis menyampaikan rasa syukur kepada Allah SWT sehingga makalah ini dapat diselesaikan tepat pada waktunya. Penulis juga menyampaikan terima kasih kepada Bapak Dr. Ir. Rinaldi Munir, M.T selaku dosen mata kuliah IF4020 Kriptografi yang telah membagikan ilmu kriptografi kepada penulis.

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 11 Maret 2020

Letivany Aldina
13514067