MODIFIKASI *VIGENERE CIPHER* DENGAN MENGGUNAKAN TEKNIK SUBSTITUSI BERULANG PADA KUNCINYA

Kukuh Nasrul Wicaksono – NIM : 13505097

Program Studi Teknik Informatika, Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
Jl. Ganesha 10, Bandung
E-mail : if15097@students.if.itb.ac.id

Abstrak

Vigenere cipher merupakan salah satu algoritma klasik yang digunakan untuk menyembunyikan pesan berupa teks dari pihak yang tidak berhak dengan menggunakan teknik substitusi dimana tiap huruf pada plainteks akan disubstitusi menjadi huruf lain berdasarkan kunci yang digunakan. Berbeda dengan Caesar cipher, vigenere cipher adalah algoritma substitusi jamak dimana suatu huruf plainteks tidak selalu disubstitusi menjadi huruf yang sama, namun disubstitusi berdasarkan kunci yang digunakan.

Kelemahan algoritma vigenere cipher muncul jika panjang kunci lebih pendek dari panjang plainteksnya sehingga terdapat perulangan kunci yang digunakan untuk mengenkripsi plainteks tersebut. Kunci yang berulang tersebut menimbulkan celah berupa jumlah pergeseran yang sama untuk setiap plainteks yang disubstitusi oleh huruf pada kunci yang sama sehingga huruf-huruf pesan atau plainteks dapat dikelompokkan berdasarkan kunci yang digunakan. Karena terdapat kelompok huruf-huruf plainteks yang disubstitusi dengan huruf kunci yang sama karena perulangan kunci, maka tiap kelompok huruf-huruf tersebut dapat dikenakan metode analisis frekuensi terhadapnya.

Pada makalah ini akan dibahas mengenai modifikasi algoritma vigenere cipher untuk mengurangi atau bahkan menghilangkan kelemahan tersebut diatas sekaligus memperkuat algoritma vigenere cipher. Modifikasi yang dilakukan pada algoritma vigenere cipher berupa modifikasi untuk membentuk kunci batu yang acak dan memiliki panjang sama dengan plainteks dari sebuah kunci asal dengan menggunakan teknik substitusi berulang pada kunci asal yang digunakan tersebut. Ide utama penulis untuk memodifikasi algoritma vigenere cipher ini pada intinya adalah dengan membangkitkan suatu kunci baru yang memiliki panjang sama dengan plainteks dan tidak memiliki perulangan *string* pada kunci tersebut. Kunci ini akan dibangkitkan dari sebuah kunci asal yang mudah untuk diingat. Metode pembangkitan kunci dari sebuah kunci asal tersebut akan dilakukan dengan menggunakan substitusi berulang pada kunci asalnya hingga mendapat sebuah kunci baru yang memiliki panjang sama dengan plainteksnya. Dengan menggunakan kunci baru acak dan memiliki panjang sama dengan plainteks yang telah terbentuk dari sebuah kunci asal, maka algoritma vigenere cipher akan menjadi lebih kuat.

***Kata kunci**: Vigenere Cipher, Caesar Cipher*, pembangkitan kunci acak, kunci asal, modifikasi, analisis kasiski, enkripsi, dekripsi, kelemahan algoritma.

1. PENDAHULUAN

Kriptografi atau yang sering dikenal dengan sebutan ilmu penyandian data adalah suatu bidang ilmu dan seni yang bertujuan untuk menjaga kerahasiaan suatu pesan yang berupa data-data dari pihak lain yang tidak berhak sehingga tidak menimbulkan kerugian. Pada masa sekarang ini, kriptografi atau ilmu penyandian data sering diklasifikasikan menjadi dua jenis yaitu kriptografi klasik dan kriptografi modern.

Kriptografi klasik sudah mulai dipelajari manusia sejak tahun 400 SM, yaitu pada zaman Yunani kuno. Kriptografi klasik ini digunakan oleh orang-orang terdahulu untuk menjaga kerahasiaan suatu pesan teks pada masa sebelum ditemukan komputer. Kriptografi klasik merupakan kriptografi yang berbasis karakter. Caesar chiper dan vigenere chiper merupakan beberapa contoh algoritma kriptografi klasik yang kita kenal saat ini. Sedangkan kriptografi modern adalah kriptografi yang berkembang pada zaman setelah ditemukan computer. Kriptografi modern beroperasi dalam bit, tidak seperti kriptografi klasik yang hanya beroperasi pada karakter. Namun pada prinsipnya, metode yang digunakan pada kriptografi modern diadopsi dari kriptografi klasik, namun dibuat sedemikian rupa sehingga lebih sulit dengan bantuan computer.

Teknik kriptografi klasik dapat dikategorikan menjadi dua bagian yaitu teknik substitusi dan teknik transposisi. Pada teknik transposisi, huruf-huruf pada plainteks hanya dimanipulasi letak posisinya (transpose) untuk menjadi teks sandi, sedangkan pada teknik substitusi, setiap huruf dalam plainteks akan tepat berkorespondensi satu-satu dengan huruf dalam teks sandinya. Untuk meningkatkan tingkat kompleksitas, improvisasi dapat dilakukan terhadap teknik-teknik tersebut.

Vigenere chiper merupakan salah satu algoritma kriptografi klasik untuk menyandikan suatu plainteks dengan menggunakan teknik substitusi. Vigenere cipher pada dasarnya cukup rumit untuk dipecahkan. Meskipun begitu, vigenere cipher tetap memiliki kelemahan. Salah satunya adalah dapat diketahui panjang kuncinya dengan menggunakan metode kasiski. Hal ini disebabkan karena terdapat frasa yang berulang-ulang pada cipherteks yang dihasilkan.

Makalah ini akan menjelaskan tentang pengembangan algoritma vigenere cipher untuk memperkuat sekaligus mengurangi kelemahan dari algoritma ini. Algoritma vigenere cipher dikembangkan dengan memodifikasi penggunaan kunci sehingga perulangan frasa yang akan dihasilkan pada cipherteks dapat dikurangi atau bahkan dihindari.

Salah satu hal yang dapat dilakukan untuk memperkuat dan mengurangi kelemahan vigenere cipher adalah dengan melakukan pengacakan ulang pada kunci yang digunakan algoritma tersebut di setiap perulangan kunci. Namun pengacakan kunci tersebut harus dilakukan dengan berpola agar penerima tetap dapat mendekripsi cipherteks yang dienkripsi dengan algoritma vigenere cipher yang dimodifikasi tersebut. Untuk dapat didekripsikan kembali, perubahan kunci pada setiap pengulangannya harus dapat memiliki parameter-parameter tertentu sehingga perubahan kunci dapat diketahui oleh penerima pesan.

Pengacakan kunci vigenere cipher dapat dilakukan dengan cara mengenkripsi kunci tersebut dengan algoritma tertentu setiap kali pengulangan kunci terjadi. Metode enkripsi pada kunci tersebut tidak harus rumit, misalnya saja dapat menggunakan algoritma caesar cipher. Dengan mengenkripsi kunci di setiap terjadi perulangan kunci tersebut, maka akan meminimalisasi kemungkinan kunci tersebut dapat dideteksi dengan penggunaan metode kasiski oleh kriptoanalis. Cara enkripsi kunci di setiap perulangan yang terjadi akan dijelaskan dengan lebih detil pada bab selanjutnya.

2. KONSEP DASAR

2.1. Caesar Cipher

Caesar cipher merupakan salah satu jenis algoritma klasik yang biasanya digunakan untuk menyembunyikan pesan. Algoritma ini sudah digunakan pada masa kekaisaran romawi. Nama Caesar sendiri digunakan sebagai nama algoritma ini karena pada masa itu algoritma ini digunakan oleh kaisar Romawi pada setiap pesan yang akan dikirimkan. Proses enkripsi pada algoritma sederhana tersebut adalah dengan mengganti karakter-karakter pada pesan yang akan dienkripsi dengan huruf ke 3 setelah huruf karakter pada pesan tersebut. Sebagai contoh karakter A pada pesan diganti dengan huruf ke 3 setelah huruf A yaitu D.

Untuk lebih jelasnya kita akan menggunakan contoh sebagai berikut.

Plainteks : CAESAR

Plainteks yang berisi pesan bertuliskan CAESAR ini akan dienkripsi dengan menggunakan metode Caesar cipher seperti yang telah dijelaskan sebelumnya yaitu dengan menggeser setiap karakter pada pesan dengan 3 huruf setelah huruf pada pesan tersebut. Hasil enkripsi pada plainteks tersebut akan menghasilkan cipherteks berikut.

Cipherteks : FDHVDU

Algoritma caesar cipher seperti yang telah dijelaskan di atas dapat dimodelkan dengan suatu fungsi matematika. Untuk proses enkripsi plainteks dapat dimodelkan sebagai berikut.

$C_i = E(P_i) = (P_i + n)$ *mod* 26

Sedangkan untuk proses dekripsi cipherteks dapat dimodelkan sebagai berikut.

$P_i = D(C_i) = (C_i - n)$ *mod* 26

Jumlah pergeseran dilambangkan dengan n. Pada algoritma caesar cipher, pergeseran yang dilakukan adalah sebanyak 3 huruf, sehingga n = 3.

Pada algoritma caesar cipher ini, setiap karakter akan disubstitusi dengan satu karakter yang sama. Karena hal tersebut algoritma ini digolongkan menjadi algoritma substitusi tunggal. Algoritma yang masuk ke dalam jenis algoritma substitusi tunggal ini mudah untuk dipecahkan oleh kriptoanalis dengan menggunakan analisis frekuensi.

Dalam makalah ini, caesar cipher tidak akan digunakan secara tunggal, namun akan digunakan bersama dengan algoritma vigenere cipher untuk dimana algoritma caesar cipher tersebut akan digunakan untuk mengenekripsi kunci dari vigenere cipher. Dengan begitu diharapkan algoritma modifikasi dari vigenere cipher tersebut menjadi lebih kuat.

2.2 Vigenere Cipher

Vigenere Cipher adalah suatu algoritma kriptografi klasik yang ditemukan oleh Giovan Battista Bellaso. Beliau menuliskan metodenya tersebut pada bukunya yang berjudul La Cifra del. Sig. Giovan Battista Bellaso pada tahun 1553. Nama vigenere sendiri diambil dari seorang yang bernama Blaise de Vigenere. Nama vigenere diambil sebagai nama algoritma ini karena beliau menemukan kunci yang lebih kuat lagi untuk algoritma ini dengan metode autokey cipher meskipun algoritma dasarnya telah ditemukan lebih dahulu oleh Giovan Battista Bellaso.

Algoritma vigenere cipher ini adalah suatu algoritma yang dirancang untuk memperbaiki kelemahan dari algoritma substitusi tunggal. Untuk diketahui pada algoritma substitusi tunggal, setiap karakter pada plainteks disubstitusikan dengan karakter yang sama. Algoritma substitusi tunggal ini mudah dipecahkan dengan menggunakan metode analisis frekuensi, yaitu dengan menghitung frekuensi kemunculan huruf tertentu dan membandingkannya dengan frekuensi kemunculan huruf pada bahasa tertentu. Misalnya pada bahasa inggris, huruf e adalah huruf yang paling sering muncul, sedangkan pada bahasa Indonesia, huruf A adalah huruf yang paling sering muncul.

Algoritma vigenere cipher dirancang untuk menghilangkan pola frekuensi huruf pada cipherteks. Dengan begitu, cipherteks yang dihasilkan dengan algoritma vigenere cipher ini tidak akan bisa dikenakan metode analisis frekuensi. Algoritma ini bekerja dengan cara yang hampir mirip dengan algoritma caesar cipher. Namun pada algoritma vigenere cipher ini, pergeseran yang dilakukan tidak selalu sama seperti pada Caesar cipher. Pada algoritma vigenere cipher, jauh pergeseran huruf plainteks ditentukan nilai desimal oleh huruf kunci yang bersesuaian dengan plainteksnya (a = 0, b = 1, c = 2, d = 3, ..., z = 25). Vigenere cipher menggunakan suatu kunci yang memiliki panjang tertentu. Panjang kunci tersebut bisa lebih pendek ataupun sama dengan panjang plainteks. Jika panjang kunci kurang dari panjang plainteks, maka kunci yang tersebut akan diulang secara periodik hingga panjang kunci tersebut sama dengan panjang plainteksnya.

Sebagai contoh, jika plainteks adalah INFORMATIKA dan kunci adalah ITB, maka proses enkripsi yang terjadi adalah sebagai berikut.

Plainteks : INFORMATIKA
Kunci : ITBITBITBIT

Cipherteks : QGGWKNIMJST

Pada contoh di atas, kata kunci ITB diulang sedemikian rupa hingga panjang kunci menjadi sama panjang dengan plainteksnya. Kemudian setelah panjang kunci menjadi sama panjang dengan panjang plainteks, proses enkripsi dilakukan dengan melakukan menggeser (seperti pada caesar cipher) setiap huruf pada plainteks sesuai dengan huruf kunci yang bersesuaian dengan huruf plainteks tersebut. Pada contoh di atas, huruf pertama pada plainteks yaitu I, akan dilakukan pergeseran huruf dengan n=8 (kunci pertama adalah huruf I yang memiliki n=8) menjadi Q, kemudian huruf plainteks kedua akan digeser juga dengan jauh pergeseran sesuai huruf kedua pada kunci, dan begitu seterusnya hingga semua plainteks telah terenkripsi menjadi cipherteks. Kemudian untuk mempersulit pemecahan cipherteks, biasanya karakter selain huruf seperti spasi, titik, koma, semicolon, dan sebagainya dihilangkan terlebih dahulu dari plainteks asal.

Jika kita amati contoh diatas, huruf I pada kata INFORMATIKA tidak disubstitusi menjadi huruf yang sama. Pada contoh tersebut huruf I pada karakter ke-1 disubstitusi dengan huruf Q, sedangkan huruf I pada karakter ke-9 disubstitusi menjadi huruf M. Dengan begitu, analisis frekuensi menjadi sulit dilakukan pada cipherteks ini sebab satu huruf yang sama pada plainteks belum tentu disubstitusi oleh karakter yang sama pula. Dapat dikatakan algoritma vigenere cipher ini lebih kuat daripada algoritma caesar cipher.

Cara untuk mendekripsikan cipherteks algoritma vigenere cipher ini hampir sama seperti cara untuk mengenkripsinya. Bedanya hanya pada arah pergeseran karakter yang dilakukan. Tiap karakter pada cipherteks akan digeser ke kiri (di substitusi dengan huruf sebelum huruf cipherteks tersebut) sesuai dengan huruf kunci yang bersesuaian dengan cipherteks tersebut.

2.2 Kelemahan Vigenere Cipher

Meskipun dapat dikatakan bahwa algoritma vigenere cipher lebih kuat daripada algoritma caesar cipher, algoritma ini tetap memiliki kelemahan sehingga cipherteks hasil dari algoritma vigenere cipher ini dapat dibuka secara paksa oleh kriptoanalis. Kelemahan ini muncul jika panjang kunci lebih pendek dari panjang plainteksnya sehingga terdapat perulangan kunci yang digunakan untuk mengenkripsi plainteks tersebut. Kunci yang berulang tersebut menimbulkan celah berupa jumlah pergeseran yang sama untuk setiap plainteks yang disubstitusi oleh huruf pada kunci yang sama. Jika kita lihat pada contoh sebelumnya huruf ke-1, 4, 7, dan 10 akan digeser dengan nilai pergeseran yang sama karena huruf-huruf tersebut memiliki kunci yang sama yaitu I karena terjadinya perulangan kunci. Hal yang sama juga terjadi pada karakter ke 2,5,8, dan seterusnya. Karena terdapat kelompok huruf-huruf plainteks yang disubstitusi dengan huruf kunci yang sama karena perulangan kunci, maka tiap kelompok huruf-huruf tersebut dapat dikenakan metode analisis frekuensi terhadapnya. Jika kita amati, maka banyaknya kelompok huruf-huruf yang digeser dengan nilai yang sama karena perulangan kunci adalah sebanyak panjang kunci tersebut.

Orang pertama yang berhasil menemukan kelemahan vigenere cipher dan memecahkan cipherteksnya adalah Friedreich Kasiski pada tahun 1863. Kasiski berpendapat bahwa jika panjang kunci dapat ditemukan, maka cipherteks dapat dipecahkan dengan menggunakan analisis frekuensi pada masing-masing kelompok cipherteks yang dihasilkan. Beliau mengusulkan suatu metode untuk mencari panjang kunci dari vigenere cipher yang sekarang biasa disebut metode Kasisiki. Metode Kasiski memanfaatkan keuntungan bahwa bahasa (misalnya bahasa inggris) tidak hanya mengandung perulangan huruf, tetapi juga mengandung perulangan pasangan huruf atau tripel huruf, seperti TH, THE, dsb. Perulangan kelompok huruf tersebut memungkinkan terjadinya kriptogram yang berulang.

Sebagai contoh adalah sebagai berikut.

Plainteks : THEBEAUTYANDTHEBEAST
Kunci : ABCABCABCABCABCABCAB
Cipherteks : TIGBFCUUAAOFTIGBFCSU

Pada contoh diatas, kata THE pertama dan kata THE kedua ternyata dienkripsi menjadi kata yang sama pula yaitu TIG. Hal ini terjadi karena

kedua kata THE tersebut dienkripsi dengan menggunakan kunci yang sama yaitu ABC. Karena kunci yang digunakan sama, maka pergeseran yang dilakukan terhadap kedua kata THE tersebut juga sama.

Dengan melihat contoh di atas maka secara intuitif kita mendapat kesimpulan bahwa jika jarak antara dua buah *string* yang berulang di dalam plainteks merupakan kelipatan dari panjang kunci,

maka *string* yang sama tersebut akan muncul menjadi kriptogram yang sama pula di dalam cipherteks. Dengan begitu, jika kita dapat menemukan perulangan *string* pada cipherteks yang telah dihasilkan melalui vigenere cipher, maka besar kemungkinan bahwa panjang kunci adalah faktor dari jarak kedua *string* yang berulang tersebut.

Jika kita melihat contoh di atas bahwa jarak antara kata THE adalah 12, maka panjang kunci adalah salah satu faktor dari 12 yaitu (1,2,3,4,6,12). Apabila kita menemukan cipherteks dengan beberapa kata yang berulang, maka panjang kunci yang mungkin adalah irisan dari faktor-faktor dari beberapa jarak perulangan *string* yang ditemukan.

Berikut ini adalah langkah-langkah dalam mencari panjang kunci vigenere cipher dengan menggunakan metode Kasiski.

1. Temukan semua kriptogram yang berulang di dalam cipherteks (pesan yang panjang biasanya mengandung beberapa kriptogram yang berulang).
2. Hitung jarak-jarak semua pasangan kriptogram yang berulang
3. Hitung semua faktor (pembagi) dari jarak-jarak yang telah ditemukan tersebut (faktor pembagi menyatakan panjang kunci yang mungkin).
4. Tentukan irisan dari himpunan faktor pembagi tersebut. Nilai yang muncul di dalam irisan menyatakan angka yang muncul pada semua faktor pembagi dari jarak-jarak tersebut . Nilai tersebut kemungkinan adalah panjang kunci. Hal ini karena *string* yang berulang dapat muncul karena bertindihan (*coincidence*).

Setelah panjang kunci diketemukan, maka kita dapat melakukan analisis frekuensi terhadap kelompok cipherteks yang dienkripsi dengan menggunakan huruf kunci yang sama seperti yang telah dijelaskan sebelumnya. Dengan menggunakan analisis frekuensi di masing-masing kelompok cipherteks tersebut, maka kriptoanalis dapat menerka kunci yang digunakan.

3. MODIFIKASI VIGENERE CIPHER

Berdasarkan tulisan sebelumnya bahwa vigenere cipher memiliki kelemahan jika panjang kunci yang digunakan kurang dari panjang plainteks. Kurang panjangnya kunci terhadap plainteks menyebabkan kunci akan diulang hingga panjang kunci tersebut sama dengan panjang plainteks. Hal ini menyebabkan kemungkinan timbulnya perulangan *string* pada cipherteks hasil enkripsi yang dapat dimanfaatkan untuk menemukan panjang kunci dan lebih lanjut dimanfaatkan untuk memecahkan cipherteks tersebut.

Hal tersebut di atas tidak akan terjadi apabila panjang kunci asal yang digunakan adalah sepanjang plainteks dan tidak ada perulangan *string* pada kunci tersebut. Hal tersebut akan sedikit rumit untuk dilakukan jika kunci yang kita gunakan merupakan suatu kata atau frasa yang memiliki arti. Namun sebaliknya, jika kunci yang digunakan bukan merupakan kata atau frasa yang memiliki arti, kunci tersebut akan menjadi sulit diingat. Dua hal inilah yang kemudian menjadi pertimbangan penulis untuk mengembangkan suatu algoritma baru yang merupakan modifikasi dari algoritma vigenere cipher tradisional agar lebih kuat dan tidak mudah dipecahkan oleh pihak yang tidak berhak.

Ide penulis untuk memodifikasi algoritma vigenere cipher ini pada intinya adalah dengan membangkitkan suatu kunci baru yang memiliki panjang sama dengan plainteks dan tidak memiliki perulangan *string* pada kunci tersebut. Kunci ini akan dibangkitkan dari sebuah kunci asal yang mudah untuk diingat. Metode pembangkitan kunci dari sebuah kunci asal tersebut akan dilakukan dengan menggunakan substitusi berulang pada kunci asalnya hingga mendapat sebuah kunci baru yang memiliki panjang sama dengan plainteksnya.

Langkah-langkah dalam membangkitkan kunci dengan panjang sama dengan plainteks dari sebuah kunci asal adalah sebagai berikut.

1. Kunci asal ditempatkan di awal kunci. (K_1)
2. *String* kunci kedua (K_2 panjang sama dengan kunci asal) dibangkitkan dari kunci asal yaitu dengan menggeser kunci asal sejauh huruf pertama kunci asal (a = 0, b = 1, c = 2, …, z = 25).
3. *String* kunci ketiga dan seterusnya (K_i) dibangkitkan dari *string* kunci sebelumnya (K_{i-1}) yang digeser sejauh n=1.
4. Jika kunci telah sampai K_{26} dan kelipatannya (kembali ke bentuk kunci asal), maka K_{26} dan kelipatannya tersebut dienkripsi kembali menggunakan vigenere cipher dengan menggunakan plainteks yang sejajar dengan kunci sebelumnya (K_{i*26-1} dan kelipatannya) sebagai kunci enkripsinya.
5. Langkah-langkah tersebut diulang hingga panjang kunci sama dengan panjang plainteks.

Setelah langkah-langkah pembangkitan kunci tersebut di atas selesai dilakukan, maka vigenere cipher siap dilakukan seperti biasa.

Berikut ini adalah contoh pembangkitan kunci yang memiliki panjang sama dengan plainteks dari sebuah kunci asal. Kunci asal yang digunakan adalah IF.

Plainteks:
INFORMATIKAINSTITUTTEKNOLOGIBAND
UNGJALANGANESHASEP**UL**UHBANDUNG

Kunci:
IFJGKHLIMJNKOLPMQNROSPTQURVSWTXU
YVZWAXBYCZDAEBFCGDHE**CQ**DRESFTG

Cipherteks:
QSOUBTLBUTNSBDIUJHKHWZGEFFBAXTKX
SIFFAIBLIZQEWIFUKSBPWXERRVZGM

Kunci yang digarisbawahi (K_{26}) pada awalnya telah kembali ke bentuk kunci asalnya (IF). Kunci tersebut kemudian dienkripsi dengan *string* plainteks yang bersesuaian dengan kunci sebelumnya (K_{25}) yaitu UL sebagai kuncinya. Hasil enkripsi tersebut (CQ) kemudian dijadikan kunci K_{26} menggantikan IF.

Setelah kunci terbentuk dengan panjang yang sama dengan panjang plainteks, maka akan dilakukan enkripsi vigenere cipher seperti biasa dan menghasilkan cipherteks seperti diatas.

Jika kita amati kunci yang telah dibangkitkan seperti contoh diatas, kita akan mendapati bahwa kunci tersebut acak dan tidak bermakna. Hal tersebut secara langsung akan menambah kekuatan algoritma vigenere cipher ini. Kunci yang acak dan memiliki panjang sama dengan plainteks tersebut membuat cipherteks yang dihasilkan juga akan terlihat lebih acak dan sangat jarang bahkan tidak memiliki perulangan *string* sama sekali. Kalaupun terjadi perulangan *string* mungkin hanya diakibatkan karena bertindihan (*coincidence*). Hal tersebut akan membuat kriptoanalis kesulitan untuk memecahkan secara paksa cipherteks dari algoritma ini. Bahkan kriptoanalis juga akan kesulitan untuk menebak kunci dan mencari panjang kunci dengan menggunakan metode kasisiki sekalipun.

Dengan menggunakan metode pembangkitan kunci seperti di atas, kunci acak yang dapat dihasilkan akan menjadi sangat jarang untuk berulang mengingat kunci tersebut tidak hanya dienkripsi dengan cipherteks biasa, namun juga dienkripsi dengan menggunakan metode vigenere cipher (pada kunci K_{26}). Hal tersebut berlaku untuk plainteks yang sangat panjang sekalipun. Perulangan kunci hanya mungkin dihasilkan oleh kebetulan semata.

Kemudian cara untuk mendekripsikan cipherteks yang dihasilkan oleh vigenere cipher yang menggunakan pembangkitan kunci seperti di atas hampir sama dengan mendekripsikan vigenere cipher biasa. Perbedaan hanya terletak pada cara untuk mendapatkan kunci vigenere cipher untuk mendekripsikannya. Pada kunci asal hingga kunci K_{25} caranya sama dengan membangkitkan kunci pada proses enkripsi. Untuk kunci K_{26} dan kelipatannya, kunci tersebut didapatkan dengan mendekripsi kunci K_{26} semula (yang kembali ke bentuk kunci asal) menggunakan algoritma vigenere cipher dengan kunci dari *string* plainteks yang bersesuaian dengan kunci sebelumnya (K_{25}). Plainteks tersebut sebelumnya didapat terlebih dahulu dari dekripsi *string*

cipherteks yang bersesuaian dengan kunci K_{25} tadi. Setelah semua kunci didapat (panjang kunci sama dengan panjang cipherteks), maka dekripsi dapat dilakukan seperti dekripsi pada vigenere cipher biasa.

4. KESIMPULAN

Berdasarkan analisis yang telah dilakukan, dapat ditarik beberapa kesimpulan terkait dengan modifikasi algoritma vigenere cipher dengan menggunakan teknik substitusi berulang pada kuncinya seperti yang telah dijelaskan sebelumnya :

- Penggunaan teknik substitusi berulang pada kunci yang akan dipakai oleh algoritma vigenere cipher jauh lebih baik dibandingkan dengan kunci biasa karena secara statistik kemunculan kunci yang berulang jauh berkurang jika dibandingkan dengan kunci biasa.
- Statistik kemunculan dan perulangan kunci akan dipengaruhi oleh panjang kunci asal yang digunakan dan juga teks pada plainteks yang bersangkutan. Hal ini dikarenakan kunci juga akan dienkripsi dengan *string* plainteks yang sejajar dengan kunci sebelumnya di setiap kunci K_{26} dan kelipatannya. Perulangan kunci tersebut memiliki probabilitas yang jauh lebih kecil jika dibandingkan dengan kunci biasa.
- Kunci acak yang terbentuk tidak terbatas panjangnya mengingat kunci tersebut tidak hanya dienkripsi dengan menggunakan Caesar cipher biasa, namun juga menggunakan vigenere cipher dengan menggunakan kunci dari *string* plainteks yang sejajar dengan kunci sebelumnya. Hal tersebut akan menambah acak kunci yang dihasilkan.
- Modifikasi algoritma vigenere cipher ini dapat memperkuat algoritma vigenere cipher jika menggunakan kunci biasa. Hal ini karena kunci yang dibangkitkan dengan teknik substitusi berulang menjadi acak dan tidak bermakna.
- Kunci yang acak dan memiliki panjang sama dengan plainteks tersebut membuat cipherteks yang dihasilkan juga akan terlihat lebih acak dan sangat jarang bahkan tidak memiliki perulangan *string* sama sekali. Kalaupun terjadi perulangan *string* mungkin hanya diakibatkan karena bertindihan (*coincidence*).
- Hal tersebut akan membuat kriptoanalis kesulitan untuk memecahkan secara paksa cipherteks dari algoritma ini. Bahkan kriptoanalis juga akan kesulitan untuk menebak kunci dan mencari panjang kunci dengan menggunakan metode kasisiki sekalipun.
- Algoritma ini tidak menjamin 100% kuat terhadap analisis kasiski meskipun kemungkinan tersebut sangat kecil.
- Saran untuk penggunaan algoritma ini adalah menggunakan kunci asal yang panjang terlebih lagi jika plainteks sangat panjang. Hal ini untuk mengurangi kemungkinan terjadinya perulangan kunci, meskipun kemungkinan tersebut sangat kecil

DAFTAR REFERENSI

[1] Munir, Rinaldi. (2004). Bahan Kuliah IF5054 Kriptografi. Departemen Teknik Informatika, Institut Teknologi Bandung.