Modifikasi Algoritma Vigenere Cipher dengan Penanganan Kunci Baru

Yollanda Sekarrini - 13512051
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
yollandasekarrini@gmail.com

***Abstract*—Algoritma Vigenere Cipher adalah algoritma kriptografi klasik yang mengenkripsi pesan menggunakan Bujursangkar Vigenere. Algoritma ini menjadi terkenal karena tingkat kesulitan yang cukup tinggi untuk dipecahkan. Namun, pada abad ke-19, Friedrich Kasiski berhasil memecahkan Vigenere Cipher dengan menggunakan metode Kasiski. Faktor yang membuat keberhasilan pemecahan cipher tersebut adalah kunci yang digunakan secara berulang. Oleh karena itu, penulis mengusulkan penanganan kunci baru pada enkripsi dan dekripsi Vigenere Cipher. Terdapat dua tahap dalam penanganan kunci tersebut. Pertama, kunci dimodifikasi sedemikian rupa sehingga berulang bukan satu kata penuh tetapi per karakter. Kedua, dilakukan penyisipan per karakter dari pesan yang akan dienkripsi.**

***Kata Kunci*—Dekripsi, Enkripsi, Kunci, Metode Kasiski, Vigenere Cipher.**

I. PENDAHULUAN

Algoritma Vigenere Cipher ditemukan oleh Giovan Battista Bellaso yang tertulis pada bukunya berjudul La Cifra del. Sig. Giovan Battista Bellaso pada tahun 1553. Sedangkan nama Vigenere berasal dari nama seorang diplomat Perancis dan kriptologis yang berhasil menemukan kunci yang lebih kuat (Auto-Key Vigenere Cipher) yakni, Blaise de Vigenere. Algoritma cipher ini menggunakan Bujursangkar Vigenere untuk melakukan enkripsi dan dekripsi.

Vigenere Cipher pada dasarnya menggunakan teknik subsitusi. Tiap karakter pada pesan yang disandikan, disubsitusi berdasarkan pada Bujursangkar Vigenere. Tiap pasangan karakter pesan dan karakter kunci yang berbeda akan menghasilkan hasil enkripsi yang berbeda tergantung posisinya pada Bujursangkar Vigenere. Tetapi, pasangan karakter pesan dan karakter kunci yang sama akan menghasilkan karakter enkripsi yang sama sehingga algoritma ini dapat dipecahkan dengan melihat frekuensi kemunculan karakter berurutan yang sama.

Algoritma kriptografi klasik ini menjadi terkenal karena mekanisme enkripsi dan dekripsi yang mudah dimengerti dan cukup sulit untuk dipecahkan bagi pemula. Algoritma ini mendapat julukan *le chiffre indéchiffrable* yang berarti "sandi yang tak terpecahkan". Walapun begitu, algoritma ini masih punya kelemahan. Hal tersebut dikarenakan pada penggunaan kunci yang berulang-ulang sehingga ketika terdapat frasa pesan yang dienkripsi menggunakan pengulangan kunci sama, hasil enkripsi akan sama dan menyebabkan panjang kunci dapat dideteksi.

Oleh karena itu, pada makalah ini penulis akan memaparkan konsep penanganan kunci baru dalam pengenkripsian pesan menggunakan algoritma Vigenere Cipher. Pengulangan kunci tidak menggunakan satu kata penuh, melainkan per karakter. Selain itu, konsep ini juga mengambil konsep Auto-Key yang menggunakan teks pesan dalam kuncinya walaupun pada makalah ini berbeda penempatan.

II. TEORI

A. Kriptografi

Kriptografi adalah ilmu dan seni dalam pengamanan pengiriman pesan dari pihak pengirim ke pihak penerima tanpa diketahui oleh pihak ketiga. Terdapat beberapa istilah dalam kriptografi, yakni enkripsi, dekripsi, cipher teks, plain teks, kunci, algoritma kriptografi dan kriptanalis.

Enkripsi adalah proses pengubahan pesan dari yang dapat dibaca menjadi pesan yang tidak dapat dibaca menggunakan kunci tertentu sehingga pihak ketiga tidak mampu melakukan eavesdropping atau pengupingan terhadap pesan yang dikirimkan oleh si pengirim ke si penerima pesan. Sedangkan dekripsi adalah proses kebalikan dari enkripsi, yakni proses pengubahan pesan dari yang tidak dapat dibaca menjadi dapat dibaca menggunakan kunci tertentu sehingga si penerima pesan dapat membaca pesan yang dikirimkan oleh si pengirim pesan.

Cipher teks adalah pesan yang telah dienkripsi sehingga pembaca tidak dapat membaca dan mengerti makna dari pesan tersebut. Sedangkan plain teks adalah pesan biasa yang dapat dibaca. Pesan plain teks yang akan dikirim, dienkripsi terlebih dahulu menjadi cipher teks kemudian didekripsi menjadi plain teks kembali.

Untuk melakukan proses enkripsi dan dekripsi sehingga pesan teks berubah dari plain teks ke cipher teks da

kembali dari cipher teks menjadi plain teks, dibutuhkan kunci. Kunci tersebut bersifat rahasia, disitulah letak kerahasiaan suatu pesan. Pada kriptografi dua macam kunci, yakni kunci simetri dan kunci nirsimetri. Kunci simetri adalah kunci enkripsi dan dekripsi yang sama dalam melakukan pengubahan teks pesan. Sedangkan kunci nirsimetri adalah kunci enkripsi dan dekripsi yang berbeda dalam melakukan pengubahan teks pesan.

Algoritma kriptografi adalah algoritma atau aturan pengubahan teks pesan dari plain teks menjadi cipher teks kemudian diubah kembali dari cipher teks menjadi plain teks. Terdapat dua jenis algoritma kriptografi, yakni algoritma kriptografi klasik dan algortima kriptografi modern. Algoritma kriptografi klasik menggunakan dua teknik ciphering, yakni subsitusi (penggantian) dan transposisi (pengubahan tempat). Contoh algoritma kriptografi klasik, yaitu Caesar Cipher, Vigenere Cipher, Playfair Cipher dan Enigma Cipher. Algortima kriptografi klasik beroperasi dalam mode karakter, sedangkan algoritma kriptografi modern beroperasi dalam mode bit. Contoh algoritma kriptografi modern, yakni Vernam Cipher dan Block Cipher.

Kriptanalis merupakan kebalikan dari kriptografer. Kriptanalis adalah orang yang memecahkan cipher teks menjadi plain teks tanpa mengetahui kunci dari pesan tersebut. Salah satu kriptanalis yang terkenal adalah Al-Kindi.

B. Algoritma Vigenere Cipher

Algoritma Vigenere Cipher adalah salah satu algoritma klasik yang menggunakan Bujursangkar Vigenere dalam melakukan enkripsi.

Gambar 2.1 Bujursangkar Vigenere
sumber:
http://upload.wikimedia.org/wikipedia/commons/thumb/9/9a/Vigen%C3%A8re_square_shading.svg/220px-Vigen%C3%A8re_square_shading.svg.png

Setiap baris pada Bujusangkar Vigenere menyatakan huruf-huruf cipher teks yang diperoleh dengan Caesar Cipher. Proses enkripsi dinyatakan pada fungsi matematika di bawah:

$$C_i = (P_i + K_i) \bmod 26$$

Sedangkan proses dekripsinya:

$$P_i = (C_i + K_i) \bmod 26$$

Keterangan:
C_i : karakter cipher teks ke-i
P_i : karakter plain teks ke-i
K_i : karakter kunci ke-i

Apabila kunci yang digunakan tidak sepanjang teks yang akan dienkripsi, maka kunci diulang secara periodik. Contoh:

Pesan : `THISPLAINTEXT`
Kunci : `sonysonysonys`

Huruf T dienkripsi dengan kunci s menjadi:

(T + s) mod 26 = L

dan seterusnya. Hasil enkripsi juga dapat dilihat menggunakan Bujusangkar Vigenere dengan melihat huruf pada perpotongan kolom karakter plain teks dan baris karakter kunci. Algoritma Vigenere Cipher menggunakan kunci simetri sehingga kunci pada proses enkripsi juga digunakan pada proses dekripsi. Pada dasarnya, algoritma Vigenere Cipher adalah Caesar Cipher dengan kunci yang berbeda-beda tiap karakternya.

Algoritma Vigenere Cipher berhasil dipecahkan oleh Freidrich Kasiski pada abad ke-19. Pemecahan tersebut dilakukan dengan melihat frekuensi kemunculan serangkaian karakter yang muncul. Metode pemecahan tersebut dikenal dengan metode Kasiski.

C. Metode Kasiski

Metode ini mengambil keuntungan dari perulangan serangkaian karakter yang kemungkinan dienkripsi oleh serangkaian karakter kunci yang sama pula seperti th dan the. Seperti pada contoh:

Plainteks :
`CRYPTO IS SHORT FOR CRYPTOGRAPHY`
Kunci :
`abcdab cd abcda bcd abcdabcdabcd`
Cipherteks :
`CSASTP KV SIQUT GQU CSASTPIUAQJB`

Kedua potongan pada plain teks `CRYPTO` di atas dienkripsi menjadi kriptogram yang sama, yakni `CSATP`. Langkah-langkah dalam memecahkan cipher teks yang dienkrip dengan Vigenere Cipher adalah:

1. Temukan kriptogram yang berulang pada cipher teks
2. Hitung jarak antar kriptogram yang berulang tersebut

3. Hitung semua faktor pembagi dari jarak tersebut
4. Dari semua jarak yang didapat dan faktor yang telah dicari, temukan irisannya (nilai yang muncul pada semua himpunan faktor atau sebagian besar).
5. Setelah didapat panjang kunci, tentukan kunci yang mungkin. Penentuan kunci bisa dilakukan dengan *exhaustive search* atau dengan menggunakan analisis frekuensi.

III. Modifikasi Algoritma Vigenere

Pada algoritma Vigenere Cipher, apabila kunci tidak sepanjang teks pesan, maka kunci akan diulang secara periodik hingga panjang kunci sama dengan panjang teks pesan. Hal tersebut yang menjadi faktor pemecahan cipher teks dari hasil enkripsi Vigenere Cipher. Sebagai contoh pada plain teks `CRYPTO IS SHORT FOR CRYPTOGRAPHY` dengan kunci `abcd`, kata `CRYPTO` diekripsi menggunakan kunci yang sama sehingga menghasilkan cipher teks yang sama. Oleh karena itu, penulis mengajukan penanganan kunci baru pada algoritma vigenere cipher sehingga menjadi lebih sulit untuk dipecahkan.

Pada penanganan kunci baru ini, terdapat dua tahap yang harus dilakukan terlebih dahulu sebelum digunakan untuk mengenkripsi plain teks. Tahap pertama, memodifikasi kunci sedemikian rupa sehingga tidak berulang satu kata penuh, tetapi berulang pada tiap karakter. Cara kerjanya, yakni mengambil satu karakter pertama dari kunci, kemudian disambung dengan dua karakter kunci, tiga karakter kunci dan seterusnya hingga menyamai panjang pesan yang akan dienkripsi. Penanganan kunci tersebut dapat diperjelas dengan contoh berikut:

Contoh 1:
Kunci : `ABCD`
menjadi,
Kunci modifikasi tahap satu : `AABABCABCD`

Contoh 2:
Kunci : `SONY`
Menjadi,
Kunci modifikasi tahap satu : `SSOSONSONY`

Jika panjang kunci yang telah dimodifikasi tidak mencukupi panjang pesan, maka kunci diulangi lagi dari pengambilan satu karakter, kemudian dua karakter dan seterusnya sampai panjang kunci sama dengan panjang pesan yang akan dienkripsi.

Setelah kunci dimodifikasi seperti yang tertera di atas, modifikasi kunci dilanjutkan ketahap kedua. Pada tahap kedua, modifikasi diambil berdasar pada variasi algoritma Vigenere Cipher yang menggunakan *Auto-Key*, yakni kunci dilanjutkan dengan pesan teks yang dienkripsi. Jika pada *Auto-Key* pesan teks yang digunakan langsung diambil semua dari karakter awal sampai karakter terakhir dikurang panjang kunci, maka pada penanganan kunci baru ini pengambilan teks pesan hanya per karakter. Setiap iterasi pengulangan satu karakter, dua karakter dan seterusnya, dilakukan penyisipan karakter dari pesan plain teks. Untuk lebih jelas, penyisipan yang dilakukan terlihat pada contoh berikut:

Contoh 1:
Plain teks :
`CRYPTO IS SHORT FOR CRYPTOGRAPHY`
Kunci : `ABCD`
Kunci modifikasi tahap satu : `AABABCABCD`
menjadi,
Kunci baru : `ACABRABCYABCDPATABOABCIABCD`

Contoh 2:
Plain teks :
`THIS PLAINTEXT`
Kunci : `SONY`
Kunci modifikasi tahap satu : `SSOSONSONY`
menjadi,
Kunci baru : `STSOHSONISONY`

Selanjutnya, proses enkripsi dan dekripsi pesan sama dengan algoritma Vigenere Cipher biasanya.

IV. Hasil Uji

Proses pengujian ini, akan dilakukan perbandingan hasil enkripsi menggunakan algoritma Vigenere Cipher biasa dengan penanganan kunci baru. Pada pengujian pertama, dipilih:

Plain teks : `THISPLAINTEXT`
Kunci : `SONYSONYSONYS`

Dengan menggunakan algoritma Vigenere Cipher biasa, akan diperoleh hasil:

Cipher teks : `LVVQHZNGFHRVL`

Sedangkan dengan menggunakan penanganan kunci baru, diperoleh:

Kunci baru : `STSOHSONISONY`
Cipher teks : `LAAGWDOVVLSKR`

Hasil di atas menunjukkan bahwa penanganan kunci baru, cipher teks yang dihasilkan masih memberikan kesan acak. Pada contoh lain, seperti:

Plain teks :
`CRYPTOISSHORTFORCRYPTOGRAPHY`
Kunci : `ABDCABCDABCDABCDABCDABCDABCD`

Dengan algoritma Vigenere Cipher yang biasa akan

memberikan hasil:

Cipher teks: CSASTPKVSIQUTGQUCSASTPIUAQJB

Dari hasil tersebut terdapat pengulangan serangkaian karakter yang merupakan celah terpecahkannya cipher teks tersebut. Tetapi, jika menggunakan penanganan kunci baru, diperoleh:

Kunci baru : ACABRABCYABCDPATABOABCIABCDS
Cipher teks: CTYQKOJUQHPTWUOKCSMPUQORBRKQ

Cipher teks yang semula terdapat pengulangan serangkaian karakter, menjadi lebih acak dan tidak terdapat pengulangan karakter. Hal tersebut disebabkan karena pengulangan kunci yang berubah. Pada algoritma Vigenere Cipher yang biasa, kunci akan diulang sehingga akan ada kemungkinan munculnya serangkaiaan karakter yang sama pada kelipatan jumlah kunci. Seperti pada contoh di atas, munculnya serangkaian karakter yang sama terjadi pada panjang pesan keempat, delapan, dua belas dan seterusnya. Tetapi dengan menggunakan penanganan kunci baru, panjang kunci menjadi bertambah besar. Seperti pada contoh, panjang kunci yang awalnya empat dengan dilakukan modifikasi tahap pertama, diperoleh panjang kunci menjadi sepuluh. Pnajng kunci tersebut dapat dicari dengan menggunakan persamaan jumlah deret aritmatika seperti berikut:

$$Sn = n/2 \{2a + (n-1)b\}$$

Keterangan :
Sn : Jumlah deret aritmatika (panjang kunci)
n : banyak suku (panjang kunci awal)
a : suku awal (bernilai 1)
b : nilai beda antar suku (bernilai 1)

persamaan tersebut dapat disederhanakan menjadi,

$$Sn = n/2 (n + 1)$$

Sehingga, jika mengesampingkan modifikasi kunci tahap kedua, modifikasi kunci pada tahap pertama telah memberikan tingkat keamanan yang lebih dibandingkan penanganan kunci algoritma Vigenere Cipher biasa.

Selanjutnya, dengan menyisipkan karakter pesan yang akan dienkripsi ke dalam kunci, pengamanan agar tidak terpecahkannya hasil enkripsi algoritma Vigenere Cipher menjadi semakin kuat. Terbukti dari penggunaan kunci pada variasi algoritma Vigenere Cipher, yakni *Auto-Key* yang memang menambah tingkat kesulitan pemecahan cipher teks. Penambahan teks pesan tersebut juga membuat pengulangan kunci menjadi kabur atau tidak kelihatan karena penyisipan dilakukan pada tiap iterasi modifikasi tahap pertama.

Dengan mengambil contoh kunci di atas, yakni sony, dilakukan perbandingan pengulangan kunci seperti yang tertera di bawah:

Kunci : **sony**sony**sony**sony

pengulangan kunci terjadi pada kelipatan empat, seusai dengan panjang kunci, sedangkan pada kunci modifikasi tahap pertama,

Kunci modifikasi tahap pertama :
ssosonsonyssosonsony**ssosonsony**ss

pengulangan kunci terjadi pada kelipatan sepuluh, sesuai dengan jumlah deret aritmatikanya, dan terakhir untuk kunci modifikasi tahap kedua,

Kunci modifikasi tahap kedua :
stsohsonisonysspsolsonasonyisnsotsone sonyxsts

pengulangan kunci tidak terjadi. Inilah keuntungan dengan menggunakan penanganan kunci baru pada enkripsi algoritma Vigenere Cipher. Metode Kasiski yang membutuhkan analisa panjang kunci menjadi pudar dan tidak dapat dipakai.

Untuk analisa lebih lanjut, dilakukanlah proses pemecahan teks pesan yang telah dienkripsi tanpa mengetahui kunci menggunakan *Vigenere Solver Online* pada situs https://f00l.de/hacking/vigenere.php. Penganalisaan ini digunakan cipher teks pada dokumen ini http://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2014-2015/Tucil2-2015.doc, yakni pada bagian kedua (Metode Kasiski).

Pada bagian kedua tersebut, terdapat teks pesan yang telah dienkripsi. Dengan menggunakan *Vigenere Solver Online* pada situs di atas dilakukanlah analisis panjang kunci. Analisis tersebut dapat dilihat pada *screenshot* di bawah:

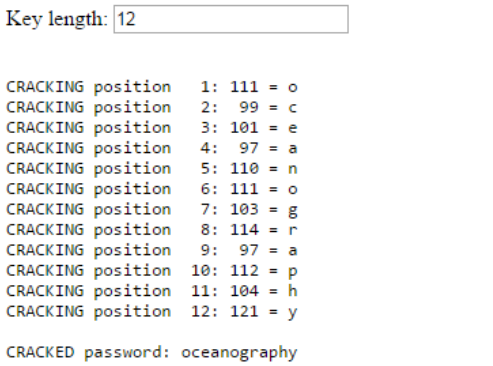
```
Checking key length of  2: min  54, max 278 ( 19.42 % ) -->  5.66 -  1.10 =  4.56 %
Checking key length of  3: min  40, max 247 ( 16.19 % ) -->  7.54 -  1.22 =  6.32 %
Checking key length of  4: min   2, max 182 (  1.10 % ) -->  7.41 -  0.08 =  7.33 %
Checking key length of  5: min  39, max 124 ( 31.45 % ) -->  6.31 -  1.99 =  4.33 %
Checking key length of  6: min   8, max 129 (  6.20 % ) -->  7.88 -  0.49 =  7.39 %
Checking key length of  7: min  26, max  89 ( 29.21 % ) -->  6.34 -  1.85 =  4.49 %
Checking key length of  8: min   2, max  87 (  2.30 % ) -->  7.09 -  0.16 =  6.92 %
Checking key length of  9: min   9, max  86 ( 10.47 % ) -->  7.88 -  0.82 =  7.06 %
Checking key length of 10: min   8, max  63 ( 12.70 % ) -->  6.41 -  0.81 =  5.60 %
Checking key length of 11: min  16, max  58 ( 27.59 % ) -->  6.50 -  1.79 =  4.70 %
Checking key length of 12: min   1, max  93 (  1.08 % ) --> 11.36 -  0.12 = 11.24 %
Checking key length of 13: min  14, max  49 ( 28.57 % ) -->  6.49 -  1.85 =  4.63 %
Checking key length of 14: min   7, max  43 ( 16.28 % ) -->  6.13 -  1.00 =  5.13 %
Checking key length of 15: min   8, max  42 ( 19.05 % ) -->  6.41 -  1.22 =  5.19 %
Checking key length of 16: min   1, max  45 (  2.22 % ) -->  7.33 -  0.16 =  7.17 %
Checking key length of 17: min  10, max  41 ( 24.39 % ) -->  7.10 -  1.73 =  5.37 %
Checking key length of 18: min   2, max  44 (  4.55 % ) -->  8.06 -  0.37 =  7.70 %
Checking key length of 19: min  10, max  37 ( 27.03 % ) -->  7.16 -  1.93 =  5.22 %
Checking key length of 20: min   1, max  48 (  2.08 % ) -->  9.77 -  0.20 =  9.57 %

The key length seems to be: 12.
```

Gambar 4.1 *Screenshot* analisis pemecahan kunci Vigenere Cipher yang biasa

Dengan menggunakan penanganan kunci yang biasanya, dapat terdeteksi bahwa panjang kunci adalah dua belas. Sebelumnya, telah dilakukan pemecahan kunci

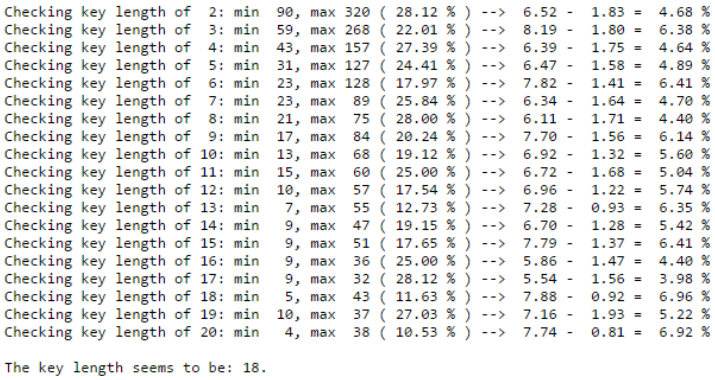
yang digunakan pada enkripsi pesan tersebut. Kunci yang digunakan adalah OCEANOGRAPHY. Hasil analisa dengan panjang kunci yang sebenarnya cocok dan setelah dilakukan *cracking*, didapat:



Gambar 4.2 *Screenshot* pemecahan kunci dengan Vigenere Cipher yang biasa

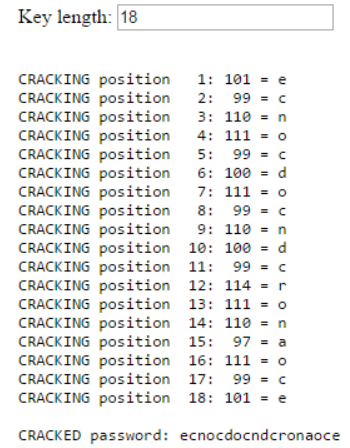
kunci yang dihasilkan adalah OCEANOGRAPHY.

Kemudian dilakukan pemecahan teks pesan yang sama tetapi dienkripsi dengan kunci yang telah dimodifikasi tahap pertama. Hasil analisa pada situs tersebut:



Gambar 4.3 *Screenshot* analisis pemecahan kunci menggunakan penanganan kunci modifikasi tahap satu

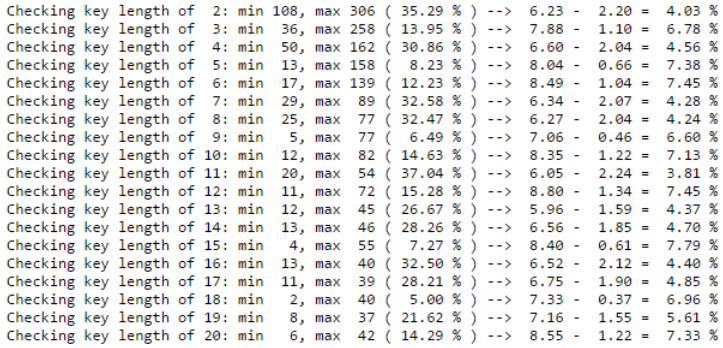
didapat hasil *cracking* sebagai berikut:



Gambar 4.4 *Screenshot* pemecahan kunci dengan penanganan kunci baru tahap satu

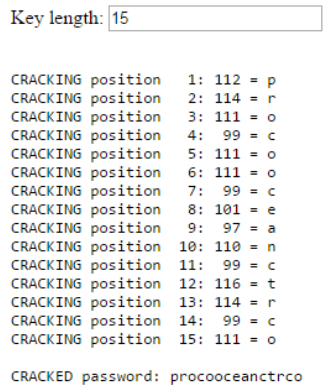
Dari hasil analisa panjang kunci dan *craking* didapat hasil yang salah. Seharusnya panjang kunci pengulangan kunci yakni n/2 (n+1) = 78. Dengan dilakukannya pengulangan per karakter seperti di atas, panjang kunci menjadi sulit untuk dianalisis dengan analisis frekuensi kemunculan.

Selanjutnya dilakukan analisis menggunakan penanganan kunci tahap kedua. Pesan teks yang digunakan masih sama, tetapi dienkripsi dengan menggunakan kunci yang telah dimodifikasi tahap kedua. Hasil analisis menunjukkan :



Gambar 4.5 *Screenshot* analisis pemecahan kunci menggunakan penanganan kunci modifikasi tahap kedua

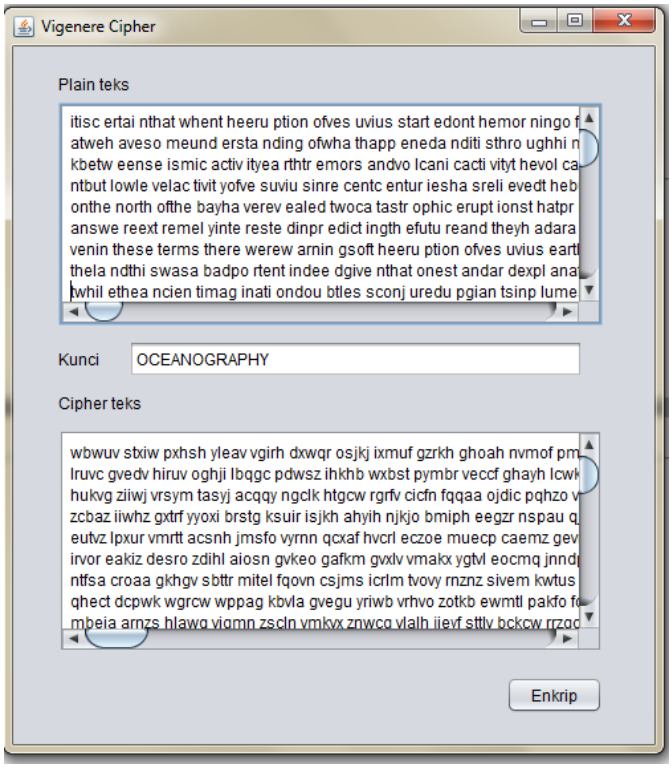
dan hasil *cracking*-nya adalah:



Gambar 4.6 *Screenshot* pemecahan kunci dengan penanganan kunci baru tahap kedua

hasil pemecahan kunci di atas semakin menunjukkan bahwa dengan menggunakan penanganan kunci baru, panjang kunci menjadi tersamarkan sehingga sulit atau bahkan tidak mungkin dilakukan penganalisisan panjang kunci dan frekuensi kemunculan serangkaian karakter. Oleh karena itu, dengan menggunakan penanganan kunci baru seperti di atas, tingkat keamanan menjadi lebih tinggi dan pemecahan algoritma Vigenere Cipher menggunakan Metode Kasiski menjadi sulit untuk dilakukan.

V. ANTARMUKA PROGRAM



Gambar 5.1 Antarmuka program

VI. KESIMPULAN

Berdasarkan hasil uji coba yang telah dilakukan, dapat disimpulkan bahwa algoritma Vigenere Cipher dapat ditingkatkan keamanannya dengan penanganan kunci baru seperti di atas. Kunci tidak diulang secara utuh, tetapi diulang secara karakter sehingga seolah-olah menambah panjang kunci. Penanganan kunci baru dilakukan sebanyak dua tahap, yakni pertama dilakukan pengulangan satu karakter kemudian disambung dua karakter dan seterusnya. Setelah itu pada tahap kedua, untuk lebih memperkuat keamanan, kunci, yang telah dimodifikasi pada tahap pertama disisipkan dengan karakter pada plain teks yang akan dienkripsi untuk setiap iterasi yang dilakukan pada tahap pertama. Selanjutnya pesan dienkripsi seperti biasa dengan menggunakan kunci yang telah dimodifikasi tersebut.

Dengan menggunakan penanganan kunci ini, metode Kasiski yang sebelumnya digunakan untuk memecahkan cipher teks pada penggunaan algoritma Vigenere Cipher menjadi lebih sulit untuk digunakan. Karena, tidak bisa menemukan panjang kunci dengan tepat dan tidak efektifnya menggunakan analisis frekuensi kemunculan serangkaian karakter yang sama.

REFERENSI

[1] http://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2014-2015/Pengantar%20Kriptografi%20(2015).ppt
Diakses pada tanggal 9 Mei 2015.

[2] http://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2014-2015/Algoritma%20Kriptografi%20Klasik_bag1%20(2015).ppt
Diakses pada tanggal 9 Mei 2015.

[3] http://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2014-2015/Algoritma%20Kriptografi%20Klasik_bag2%20(2015).ppt
Diakses pada tanggal 9 Mei 2015.

[4] http://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2014-2015/Kriptanalisis%20(2015).ppt
Diakses pada tanggal 9 Mei 2015.

[5] https://f00l.de/hacking/vigenere.php
Diakses pada tanggal 10 Mei 2015

PERNYATAAN

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, 10 Mei 2015

ttd

Yollanda Sekarrini (13512051)