KOMBINASI ALGORITMA RUBIK, CPSRNG CHAOS, DAN S-BOX FUNGSI LINIER DALAM PERANCANGAN KRIPTOGRAFI CIPHER BLOK

Vania Beatrice Liwandouw[1)], Alz Danny Wowor[2)]
Program Studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Kristen Satya Wacana
Jl. Diponegoro 50-66, Salatiga, 97751
Telp : (0298) 3419240, Fax : (0298) 3419240
E-mail : 672012224@student.uksw.edu[1)] alzdanny.wowor@staff.uksw.edu[2)]

Abstrak

Pengamanan data dalam berkomunikasi menjadi hal yang penting. Kriptogafi cipher block sering digunakan sebagai alat pengamanan. Penelitian ini merancang algoritma cipher block dengan mengkombinasikan algoritma rubik, CPSRNG berbasis chaos, dan juga s-box dengan fungsi linier. Hasil penelitian ini diperoleh bahwa kombinasi algoritma terbukti sangat baik karena menunjukkan korelasi yang melemah atau mendekati nol sehingga plainteks dan cipherteks tidak ditemukan hubungannya. Ruang kunci yang cukup besar membuat algoritma ini sangat kuat terhadap serangan kriptanalisis brute-force attack.

Kata kunci: *Rubik, CPSNRG Chaos, S-Box, Linear Functions, Cipher Block, Cryptography.*

Abstract

Cryptography is an important requirement in securing the data and information. Cryptography block ciphers are frequently used as a security tool. This paper designed a block cipher algorithm by combining a Rubik algorithm, based CPSRNG chaos, and also s-box with a linear function. The result of the research showed that the combination of the algorithm proved are very good because it shows the correlation is weakened or close to zero so that plaintext and ciphertext have no relation. A large enough key space make the algorithm is very strong to against brute-force attack.

Keywords: *Rubik, CPSNRG Chaos, S-Box, Linear Functions, Cipher Block, Cryptography.*

1. PENDAHULUAN

Kriptografi blok cipher sering digunakan sebagai pengamanan data dalam pengiriman dan atau pertukaran informasi, teknik ini banyak digunakan kerena kebutuhan proses yang lebih efisien dalam komputer digital khususnya kebutuhan waktu dan memori. Selain itu rancangan algoritma blok cipher dapat diimplementasikan di
berbagai *platform* [1]. Terkait dengan cipher blok, banyak algoritma yang digunakan untuk merancang sebuah kriptografi. Berdasarkan penelitian Liwandouw & Wowor [2], penggunaan rubik untuk mendesain sebuah algoritma kriptografi simetris dengan jenis *cipher* dapat mengakomodasi proses transposisi yang unik terkait pemasukan dan pengambilan bit sehingga dapat memiliki tingkat keacakan yang baik dan mampu menghilangkan korespodensi yang linier antara plaintek dan cipherteks.

Mengetahui tingkat keacakan pada sebuah teknik kriptografi dibutuhkan untuk melihat korespondensi satu-satu antara plainteks dan cipherteks, pada kondisi tersebut apabila relasi plainteks-cipherteks berpola maka secara statistika akan mudah dipecahkan oleh kriptanalisis. Namun, tingkat keacakan saja belum cukup kuat untuk sebuah kriptosistem, karena pada kondisi tertentu dengan inputan plainteks dengan bit nol atau satu semua maka cipherteks juga akan menghasilkan bit yang sama dengan plainteks yaitu nol atau satu semua. Hal ini terjadi apabila rancangan kriptografi hanya memperhatikan proses transposisi saja. Kelemahan ini juga diperhatikan oleh kriptografi DES, AES, GOST, dan lainnya. Untuk menghindari serangan kriptanalisis karena kelemahan algoritma dengan sebuah kotak subtitusi (*s-box*) yang menghilangkan hubungan yang berpola antara plainteks-cipherteks.

Hal ini yang kemudian menjadi ide dasar dalam penelitian ini yaitu dengan mengkombinasikan rancangan algoritma rubik dengan CPSNRG (*cryptographically secure pseudorandom generator*) berbasis chaos dan *S-Box* dengan fungsi linier.

Pemilihan CPSNRG Chaos dikarenakan perlunya bilangan acak yang tidak dapat diprediksi dan tidak memiliki periode perulangan. Sedangkan Rancangan S-box menggunakan fungsi linier karena fungsi linier memiliki invers yang dibutuhkan untuk proses dekripsi.

Penelitian ini merancang sebuah kriptografi blok cipher menggunakan beberapa motode yang dirancang dan kemudian dikombinasikan menjadi sebuah sistem untuk mengamankan informasi berupa teks.

2. KAJIAN PUSTAKA

2.1 CPSNRG Berbasis Chaos

CPSNRG atau *cryptographically secure pseudorandom generator* merupakan pembangkit bilangan acak yang dapat menghasilkan bilangan yang tidak dapat diprediksi [3]. Chaos ditemukan oleh Edward Lorentz pada tahun 1960 yang digunakan untuk membuat model perkiraan cuaca, model tersebut diberikan pada Persamaan (1) yang dilanjutkan sebagai model iterasi pada Persamaan (2).

$$f(x) = rx(1-x) \tag{1}$$

yang dapat dinyatakan dalam bentuk iteratif sehingga menjadi

$$x_{i+1} = rx_i(1-x_i) \tag{2}$$

Chaos dipakai sebagai CPSNRG karena memiliki efek kupu-kupu (*butterfly effect*) karena perubahan kecil pada nilai inputan berakibat terjadi perubahan yang sangat signifikan pada nilai output [3].

2.2 S-Box

Proses subtitusi yang memetakan inputan berdasarkan *look-up table*. Biasanya imputan dari operasi pada *s-box* dijadikan indeks dan keluaran adalah entrinya. Terdapat empat pendekatan yang dapat digunakan untuk perancangan *s-box* [3]. Dipilih secara acak, dipilih secara acak dan diuji kembali, teknik *man-made*, dan cara *math-made*. Penelitian ini merancang *s-box* dengan fungsi linier

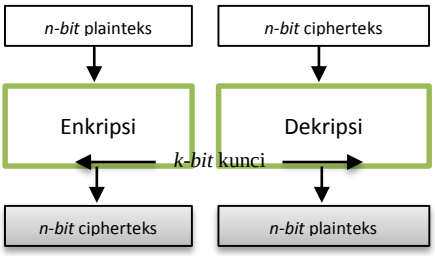
$$g(x) = ax + b \tag{3}$$

Untuk perancagan dari invers *s-box* digunakan invers dari fungsi linier yang secara umum diberikan pada Persamaan (2).

$$g^{-1}(x) = (x-b)/a \tag{4}$$

2.3 Blok *Cipher*

Cipher blok merupakan rangkaian bit yang dibagi menjadi blok-blok bit dengan panjang yang sama. Proses Enkripsi dilakukan terhadap blok bit plainteks yang ukurannya sama dengan ukuran blok plainteks [3]. Skema untuk proses enkripsi dan dekripsi ditunjukkan pada Gambar 1.



Gambar 1. Skema enkripsi dan dekripsi cipher blok [4]

Misalkan blok plainteks dan cipherteks berukuran *n*-bit dinyatakan sebagai $P = (p_1, p_2, ..., p_n)$ dimana p_i untuk $i = 1, 2, ..., n$, dan $C = (c_1, c_2, ..., c_n)$ dimana c_i untuk $i = 1, 2, ..., n$. Proses enkripsi dan dekripsi dengan kunci *K* dinyatakan berturut-turut dengan Persamaan (5).

$$E_K(P) = C \;\; ; \;\; D_K(C) = P \tag{5}$$

2.4 Rubik

Kubus Rubik (*Rubik's Cube*) jika masih belum diacak, mainan yang berbentuk kubus berwarna ini hanya terlihat seperti kotak biasa yang tidak menyenangkan untuk dimainkan. Namun jika diacak, mainan yang memiliki sembilan petak di setiap sisinya ini merupakan *puzzle* yang cukup sulit untuk diselesaikan dalam waktu singkat bagi orang yang awam dalam memainkannya. Bahkan terkadang, mainan ini dapat menyebabkan sakit kepala jika orang tersebut tidak tahu bagaimana cara menyelesaikan kubus berwarna-warni tersebut.

Mengacak warna di mainan ini sangat mudah, yang sulit adalah cara untuk mengembalikan warna setiap sisi agar kembali seperti semula. Cara memainkannya hanya dengan memutar setiap bagian dari kubus tersebut, baik secara vertikal maupun horizontal. Tujuannya adalah membuat masing-masing sisi sesuai dengan warna semula sebelum mainan tersebut diacak.

Ide yang mendasari mainan ini ditemukan oleh seorang profesor arsitektur berkebangsaan Hungaria bernama Ernő Rubik pada tahun 1970-an. Kepopuleran mainan ini melejit pada awal tahun '80-an pada saat mainan ini diberitakan di majalah Omni pada tahun 1980. Sebanyak 300 juta Kubus Rubik dijual pada saat itu [5].

2.5 Sistem Kriptografi

Stinson [6], menjelaskan sebuah sistem kriptografi harus memenuhi lima-tuple (*five-tuple*) yang terdiri dari (***P***, ***C***, ***K***, ***E***, ***D***) dimana: ***P*** adalah himpunan berhingga dari plainteks, ***C*** adalah himpunan berhingga dari cipherteks, ***K*** merupakan ruang kunci (*keyspace*), adalah himpunan berhingga dari kunci. Untuk setiap $k \in \boldsymbol{K}$, terdapat aturan enkripsi $e_k \in \boldsymbol{E}$ dan berkorespodensi dengan aturan dekripsi $d_k \in \boldsymbol{D}$. Setiap $e_k : \boldsymbol{P} \rightarrow \boldsymbol{C}$ dan $d_k : \boldsymbol{C} \rightarrow \boldsymbol{P}$ adalah fungsi sedemikian hingga $d_k(e_k(x)) = x$ untuk setiap plainteks $x \in \boldsymbol{P}$.

2.6 Korelasi

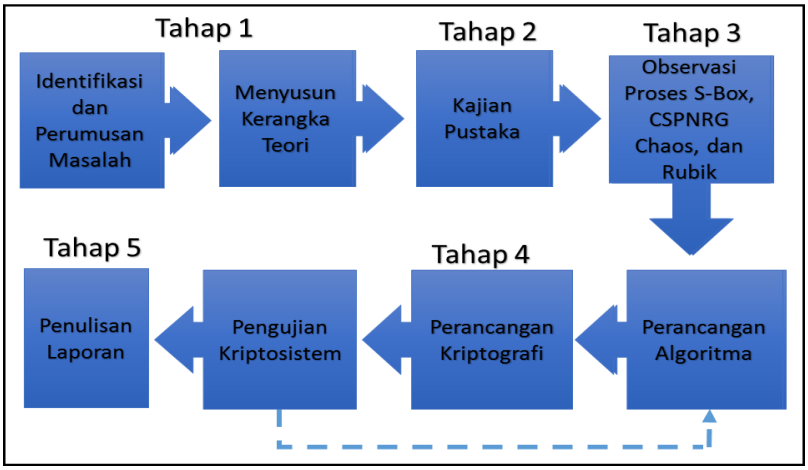
Analisis korelasi digunakan untuk melihat hubungan secara linier antara dua peubah yang biasanya adalah *X* dan *Y*, secara umum diberikan pada Persamaan (6) [7].

$$r = \frac{n\sum xy - (\sum x)(\sum y)}{\sqrt{\{n\sum x^2 - (\sum x)^2\}\{n\sum y^2 - (\sum y)^2\}}}, \tag{6}$$

Dimana *n* adalah banyaknya karakter, $\sum x$ adalah total jumlah dari variabel *x* (bilangan ASCII plainteks), $\sum y$ adalah total jumlah dari variabel *y* (bilangan ASCII cipherteks), Σx^2 adalah kuadrat dari total jumlah variable *x*, Σy^2 adalah kuadrat dari total jumlah variabel *y*, Σxy adalah hasil perkalian dari total jumlah variabel *x* dan variabel *y*.

3. METODE PENELITIAN

Bagian ini membahas tentang langkah-langkah (tahapan) yang dilakukan untuk menyelesaikan permasalahan penelitian. Secara lengkap tahapan penelitian diberikan pada Gambar berikut:

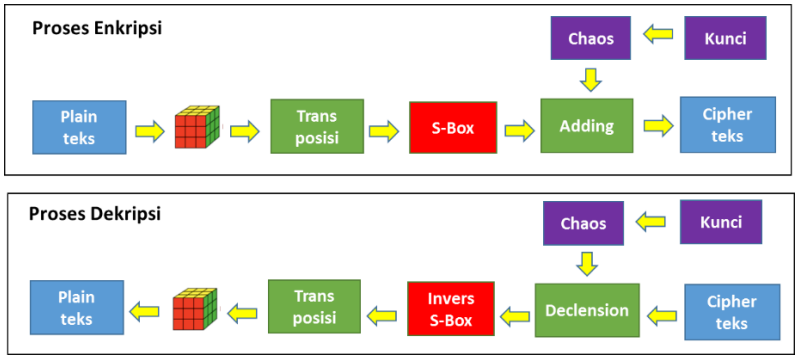


Gambar 1

Penjelasan lengkap terkait dengan langkah-langkah (tahapan) yang telah dan akan dilakukan beserta juga dengan hasil (*output*) yang sudah/akan diperoleh diberikan pada tabel berikut ini.

Tahapan	Aktifitas	Output
Tahap 1	Mengidentifikasi dan merumuskan masalah melalui kajian pustaka yang bersumber pada buku, jurnal yang relevan.	Rumusan permasalahan yaitu bagaimana membuat kriptografi yang berbasis bagaimana merancang algoritma berbasis pada Rubik
	Menyusun kerangka teori terkait dengan masalah yang telah dirumuskan.	Memperoleh suatu rancangan kerangka teori yang telah disesuaikan dengan rubik.
Tahap 2	Kajian Pustaka	Memperoleh pustaka, baik dari buku, jurnal maupun narasumber yang mengetahui tentang kriptografi berbasis rubik
Tahap 3	Observasi Proses *S-Box*, CSPNRG Chaos, dan rubik	Mengetahui cara kerja *s-box*, CPSNRG chaos, dan kubus rubik 4 × 4 × 4
Tahap 4	Perancangan Algoritma	Menghasilkan algoritma
	Perancangan Kriptografi	Menghasilkan kriptografi
	Pengujian Kriptosistem	Mengetahui kekuatan kriptografi yang telah dirancang serta menghasilkan sebuah sistem kriptografi yang telah memenuhi aturan Stinson.
Tahap 5	Penulisan Laporan	Menghasilkan laporan penelitian dalam bentuk jurnal.

Setiap langkah-langkah atau tahapan secara jelas telah ditunjukkan pada tabel diatas. Tetapi yang menjadi catatan pada Langkah 4 adalah pengujian sebuah sistem kriptografi berdasarkan aturan Stinson. Rancangan kriptografi berbasis pada rubik memiliki begitu banyak proses yang perlu dilakukan, dalam hal ini ruang plainteks, ruang kunci, ruang cipherteks dan juga proses enkripsi maupun dekripsi. Setelah algortima berhasil dirancang, maka selanjutnya perlu dilakukan pengujian 5-*tuple* untuk memenuhi sebuah kriptosistem. Adapun rancangan kriptografi secara umum dijelaskan pada Gambar 2.



Gambar 2. Proses Enkripsi dan Dekripsi

Pada proses enkripsi, *n*-plainteks sebagai input dikenakan proses rubik, dimana plainteks yang telah dirubah menjadi blok bit diacak secara vertikal maupun horizontal. Kemudian terjadi proses transposisi blok bit. Setelah

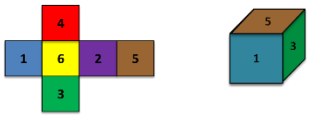
itu dikenakan proses S-box dengan kombinasi kunci dan pembangkitan bilangan acak berbasis *Chaos* yang kemudian menghasilkan cipherteks. Sedangkan pada proses dekripsi, berlaku sebaliknya.

4. RANCANGAN KRIPTOGRAFI

Berdasarkan rancangan kriptografi yang telah diberikan pada Gambar 2, bagian pertama yang akan dibahas adalah algoritma rubik, kemudian secara berturut akan dibahas rancangan *s-box* dengan fungsi linier, dan pembentukan CPSRNG yang berbasis chaos.

4.1 Algoritma Rubik

Rubik yang digunakan adalah kubus rubik 4×4×4, yang memiliki 64 kubus kecil dimana setiap kubus kecil mempunyai 6 sisi seperti pada Gambar 3. Setiap sisi dari rubik tersebut dirancang untuk ditempatkan sebuah bit sehingga pada setiap kubus kecil akan ditempati 6 bit, oleh karena itu secara total sebuah rubik 4×4×4 akan menampung 384 bit.



Gambar 3. Enam sisi pada Rubik Kecil [2]

Gambar 4. Proses Akhir Rubik [2]

Penggunaan rubik dengan asumsi akan ditempatkan bit kesetiap sisi pada rubik, dan kemudian rubik tersebut akan di putar sehingga akan memposisikan bit pada tempat/posisi yang berbeda dan kemudian akan diambil kembali bit yang teracak. Sebagai contoh setiap bit disusun dan diposisikan ke dalam kubus rubik secara horizontal seperti pada Gambar 4. Kemudian dilakukan pengambilan bit dari rubik sehingga kembali membentuk blok bit yang baru.

4.1 Rancangan *S-Box* Fungsi Linier

Penggunaan fungsi linier digunakan dalam *s-box* karena secara kalkulus sudah tentu mempunyai invers. Kondisi ini menguntungkan untuk memilih sembarang fungsi yang dapat digunakan, asalkan setiap fungsi mempunyai invers terhadap modulus 256. Pemilihan modulus 256 karena disesuaikan dengan karakter ASCII.

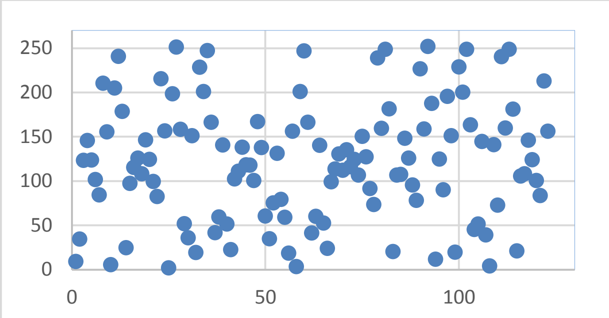
x+2	x+3	5x+2	3x+11	5x-1	7x-8	9x-5	x+6	x-1	3x-1	5x-1	x+4
4x+9	3x+7	x+5	8x+1	5x-3	11x+27	8x+3	4x-1	x+3	9x-8	4x+9	3x+7
5x+4	6x+1	x+7	9x+2	7x-2	5x-4	2x+7	6x-1	2x+9	8x-7	7x-1	x+8
8x+5	7x+6	x+20	x+10	4x-1	9x-2	x+9	3x+1	8x-3	2x+9	9x-1	2x+5

Gambar 5. S-Box Fungsi Linier

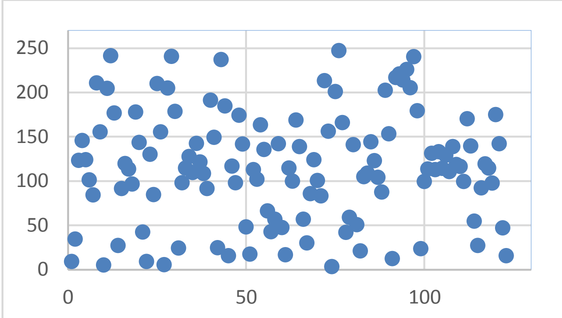
Kebutuhan ini diperlukan untuk dalam proses dekripsi. Proses untuk menentukan invers fungsi dapat mengacu pada Persamaan (4). Sebagai contoh fungsi linier pada baris dua dan kolom enam $f(x) = 11x + 27$, maka inversnya diperoleh $f^{-1}(x) = (x–27)/11$.

4.2 Pembangkitan CPSRNG *Chaos*

Proses pembangkitan dilakukan dengan mengambil inputan dari kunci. Karakter kunci di dekode dengan ASCII kemudian rata-ratakan untuk mendapatkan sebuah nilai yang digunakan sebagai konstanta pengali untuk mendapatkan nilai x_0 pada Persamaan (2).



Gambar 6. $r = 3,371113$; $x_0 = 0,0022532028$



Gambar 7. $r = 3,71114$; $x_0 = 0,0022532028$

Pembangkitan bilangan acak dengan CPSRNG yang berbasis pada chaos, digunakan Persamaan (2), dengan mengambil konstanta $r =$ dan konstanta $x_0 =$sebangai inputan nilai awal yang berbeda maka

4.4 Proses Enkripsi-Dekripsi

Plainteks diinput sebanyak n karakter dimana $n|144$; $n \in Z^+$ kemudian di dekode ke ASCII yang terkonversi ke dalam blok satu blok berukuran 1152 bit. Ukuran plainteks secara umum dapat dinotasikan menjadi

$$P = \{p_1, p_2, ..., p_n\} \quad (7)$$

dimana setiap blok bit (p_i); $i = 1, 2, ..., n$ berukuran 1152 bit.

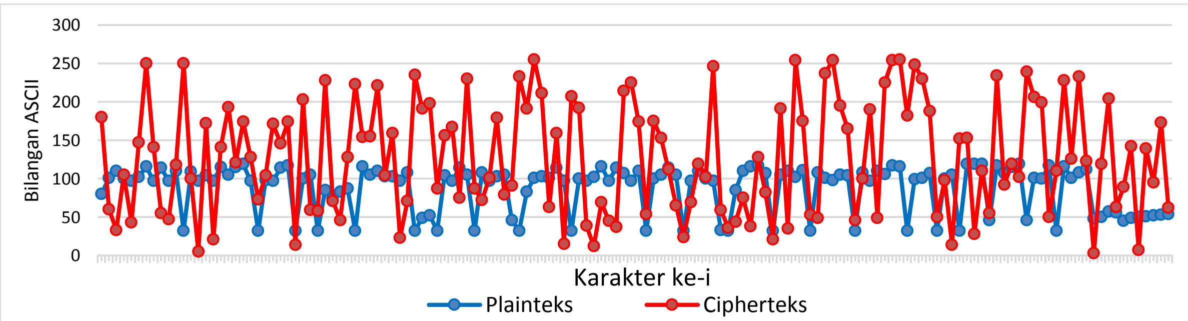
$$p_1 = \{x_1, x_2, ..., x_{1152}\},$$
$$p_2 = \{x_{1153}, x_{1154}, ..., x_{3204}\},$$
$$\dots \quad \dots$$
$$p_n = \{x_{1152n-1151}, x_{1152n-1150}, ..., x_{1152n}\}. \quad (8)$$

5. ANALISA RANCANGAN KRIPTOGRAFI

5.1 Analisa Proses Enkripsi-Dekripsi

Untuk menujukkan proses enkripsi dan dekripsi maka dilakukan dengan mengenkripsi dua buah bentuk pesan yang berbeda. Pesan pertama lebih dilihat pada perbedaan karakter atau kombinasi karakter yang berupa angka, simbol dan abjad, sedangkan pesan yang kedua adalah pesan dengan karakter yang sama. Untuk pesan yang pertama dipilih :

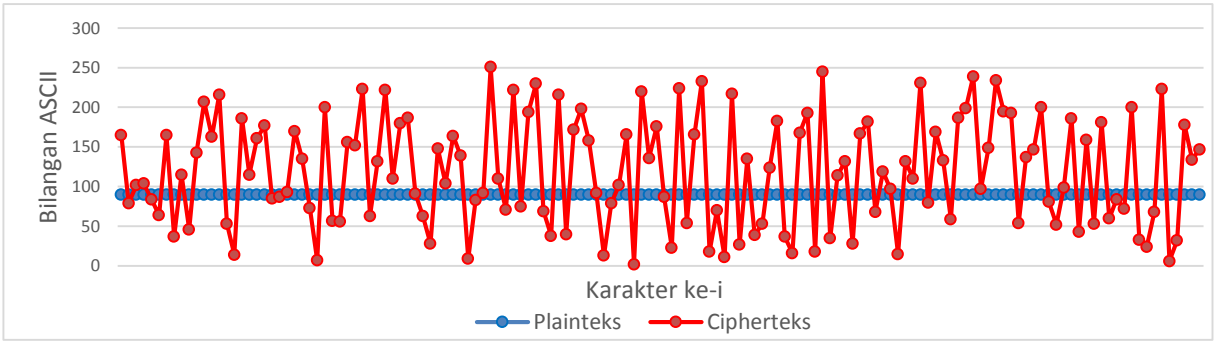
Plainteks: “Pendaftaran mahasiswa baru di UKSW tinggal 14 hari lagi. Segera daftarkan diri anda! Untuk info lebih lanjut cek di www.uksw.edu telp0298-123456” dan kunci “FTI SALATIGA” maka menghasilkan cipherteks: ´d’›<®Ý!¬hiËŸ+□;“Á:Gúyäë□®G¿7€.Æ/I€Wvhßœú«š§K?®$æŸ6,W¯KHÏ™&eÀp€³'ARO[EE¿é-w#¿%fþÿÖ ö ¯Óá;51¶ê{íø\þæwwÃ¼fÌ¥2ï?.bÎYdÇŽ¾˜21™null‹áä_þn~ÿ7é>.
Hasil dari pesan pertama scara visual ditunjukkan pada Gambar 8



Gambar 8. Contoh Pertama (Plainteks Berfariasi)

Untuk pesan yang kedua dipilih plainteks dengan karakter yang sama: ZZZZZZZZZZZZZZZZZZZZZZZ ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ dengan kunci yang sama dengan pesan satu yaitu “FTI SALATIGA” maka diperoleh cipherteks: “¥£‡nOØI´f5»hÈ[Tº9?@s8¥¡œ”%± ˜hsUß¤.W

?‹□]„Ïª÷S\¬W'ûÆ5nullžà|G\6·Þ¦%KOéÂf¨æ¦FÁE&ÜÙõØˆ#(°‡r„©‰T… “H§;ÈÈ¶»Q!DÇ4wïcDaaºß•+„êŸnÃ 5²çÁµ†P6<“. Gambar 9 menunjukkan grafik dari plainteks dan cipherteks terhadap pesan kedua. Nampak bahwa walaupun plainteks yang sama, tetapi cipherteks yang diperoleh tetap fluktuatif. Hal ini menjukkan bahwa algoritma dapat membuat plainteks menjadi sangat acak.



Gambar 9. Contoh Kedua Plainteks Karakter Sama

5.2 Analisis Korelasi

Kriptografi mengubah plainteks menjadi cipherteks, setiap algoritma yang di buat berusaha untuk menghilangkan hubungan secara langsung antara plainteks dan cipherteks agar kriptanalisis sulit untuk menentukan hubungan secara linier atau dengan teknik kriptanalis lainnya menetukan plainteks walaupun tidak mengetahui kunci. Hubungan yang unik antara plainteks dan cipherteks dapat diuji dengan melihat hubungan secara statistik. Hubungan ini plainteks dan cipherteks ditentukan dengan menggunakan analisis korelasi yang diberikan pada Persamaan (6).

Nilai korelasi pada Plainteks bervariasi yang diberikan pada Gambar 8 adalah -0.000253, sedangkan untuk plainteks kedua yang ditujukkan pada Gambar 9 adalah 0. Kedua plainteks yang dipilih mewakili jenis variasi plainteks yang mungkin, oleh karena itu dihitung nilai korelasi dari kedua jenis plainteks tersebut. Hasil korelasi dari kedua plainteks menunjukkan bahwa plainteks dan cipherteks berkorelasi sangat lemah. Analisis ini menggambarkan algoritma yang dirancang mampu untuk menghilangkan hubungan secara statistik antara plainteks dan cipherteks.

5.3 Analisis Ruang Kunci

Kriptanalisis dengan teknik *brute-force attack* akan mencoba semua kemungkinan kunci mendekripsi cipherteks. Secara teoritis agar *brute-force attack* menjadi tidak efisien dilakukan, maka jumlah kemungkinan kunci harus dibuat besar. Kunci yang dibangkitkan dengan CPSNRG Chaos sebanyak 147 kunci dinamis tergantung pada desimal dari karakter inputan kunci. Ruang kunci menyatakan jumlah total kunci yang berbeda yang dapat digunakan untuk enkripsi/dekripsi. Sehingga banyaknya nilai kemungkinan adalah 256^{144}, dimana 256 merupakan banyaknya kemungkinan dari bilangan ASCII dan 144 merupakan banyaknya karakter yang menjadi input dalam satu kali proses. Banyak kemungkinan ruang kunci yang diperoleh, dengan asumsi komputer yang tercepat saat ini dapat memecahkan sebanyak 1juta kunci, maka waktu yang dibutuhkan sebanyak $3{,}23296301 \times 10^{337}$ tahun. Kondisi ini, membuat ruang kunci terhitung cukup besar untuk dapat bertahan terhadap serangan kriptanalisis *brute-force attack*.

6. SIMPULAN

Algoritma ini dapat mengenkripsi pesan teks. Kombinasi algoritma rubik, CPSNRG chaos dan s-box fungsi linier terbukti sangat baik dan dapat menghilangkan hubungan secara statistik antara plainteks dan cipherteks. Hasil ini ditunjukkan dengan korelasi yang melemah (mendekati atau sama dengan) nol sehingga algortima sangat baik dalam menghilangkan korespodensi plainteks terhadap cipherteks. Ruang kunci yang cukup besar membuat algoritma ini sangat kuat terhadap serangan kriptanalisis *brute-force attack*.

7. DAFTAR RUJUKAN

[1] Redman, P., 2006. *Good essay writing: a social sciences guide*. 3rd ed. London: Open University in assoc. with Sage.

[2] Liwandouw, V. B., & Wowor, A.D., 2015, Desain Algoritma Berbasis Kubis Rubik dalam Perancangan Kriptografi Simetris, *Seminar Teknik Informatika dan Sistem Informasi* (*SETISI*), 9 April 2015, Bandung: Fakultas Teknologi Informasi - Universitas Kristen Marantha.

[3] Munir, Rinaldi, 2006, *Kriptografi*, Bandung: Informatika.

[4] Forouzan, Behrouz, A., 2008, *Cryptography and Network Security*, New York: Mc Graw Hill

[5] V-CUBE™ Verdes Innovations S.A. Official Web Page, https://www.v-cubes.com/products/v-classics (Diakses pada tanggal 28 Februari 2014).

[6] Stinson, D.R., 1995, *Cryptography Theory and Practice*, Florida: CRC Press, Inc.

[7] Montgomery, D.C., & Runger, G.C., 2011, *Applied Statistics and Probability for Engineers*, New York: Fifth Edition, John Wiley & Sons.

[8] Boughton, J.M., 2002. The Bretton Woods proposal: an in depth look. *Political Science Quarterly*, 42 (6), pp.564-78.

[9] Slapper, G., 2005. Corporate manslaughter: new issues for lawyers. *The Times*, 3 Sep. p. 4b.