Jurnal Ilmiah
ILMU KOMPUTER
Universitas Udayana

Vol. XI, No. 1, April 2018 **ISSN 1979 - 5661**

PENERAPAN KONSEP *SOMATIC HYPERMUTATION* DALAM ALGORITMA ENKRIPSI *ONE-TIME PAD*

Aldo Adrian[1], Ketut Bayu Yogha Bintoro[2]

Program Studi Teknik Informatika Universitas Trilogi
e-Mail: barnaa9@gmail.com[1], ketutbayu@universitas-trilogi.ac.id[2]

ABSTRAK

Aspek privasi dan keamanan di era teknologi seperti sekarang ini adalah suatu kewajiban yang harus dipenuhi oleh tiap individu. Banyak hal yang dapat mengganggu privasi dan keamanan dalam dunia teknologi informasi seperti penyadapan atau bahkan karena kurangnya rasa waspada. Kriptografi adalah salah satu cara untuk mengamankan data rahasia dengan melakukan penyandian dan mengubah isi pesan sebelum dikirim. One-Time Pad adalah suatu algoritma kriptografi yang mudah dimengerti tetapi sukar untuk diserang atau disadap. Oleh karena itu, algoritma ini dijadikan sebagai dasar pengembangannya yang menerapkan suatu konsep menarik yaitu Somatic Hypermutation dari ranah Artificial Intelligent. Penerapan konsep unik ini akan membuat suatu langkah baru di dalam algoritma One-Time Pad yang membuat keamanan dari pesan yang disandikan meningkat.

Kata Kunci: Kriptografi, *One-Time Pad, Somatic Hypermutation*.

ABSTRACT

Privacy and security in technology nowadays are one of needs that each person have to earn. There are many ways that capable to disturb these aspects, especially in information technology field, such as being tapped or even lack of awareness from the first person. Cryptography is one of the method to secure privacy data through encryption process right before it posted or sent. One-Time Pad is a cryptography algorithm which is easy to understand but near-impossible for being tapped and attacked, that's why this algorithm used for improving itself by adding a unique concept called Somatic Hypermutation from Artificial Intelligent field of study. This unique concept will add a new step into One-Time Pad algorithm that increase its encryption security level.

Keywords: *Cryptography, One-Time Pad, Somatic Hypermutation*

1. PENDAHULUAN

Pfleeger, et al (2015) menyatakan bahwa *information privacy* (privasi informasi) memiliki tiga aspek yang saling bergandengan erat, yaitu; data yang sensitif, subjek yang bersangkutan, dan pengendalian penyebaran informasinya. Ditekankan bahwa privasi adalah suatu hak bagi semua individu untuk mengontrol siapa yang mengetahui hal-hal mengenai dirinya, informasi-informasi sensitif seperti identitas, keuangan, pendidikan, dan yang lainnya. (Pfleeger, P. C., et al, 2015)

Memang seperti yang Pfleeger, dkk (2015) tekankan bahwa apa yang seseorang

anggap sebagai privasi adalah keputusannya sendiri: tidak ada standar universal mengenai hal-hal apa saja yang privasi atau pun tidak. Tetapi, Supriyatna (2014) menegaskan dari sudut pandang teknologi bahwa aspek keamanan benar-benar penting, terutama ketika seorang pengguna teknologi informasi memiliki suatu *file* rahasia penting atau *password* yang memerlukan keamanan yang tinggi. Pernyataan Supriyatna tersebut didukung kembali oleh Pfleeger, dkk (2015) yang berkata bahwa proses kecepatan tinggi komputer memungkinkan suatu pemrosesan dan kemampuan transmisi untuk mengoleksi dan mengorelasi data yang dapat berdampak negatif terhadap privasi seseorang.

Demi menjawab kebutuhan privasi di dalam bidang teknologi informasi tersebut, penulis mengemukakan suatu algoritma atau metode pengenkripsian baru yang memiliki kemampuan untuk mengenkripsi dengan keamanan yang terjamin. Algoritma ini menggunakan algoritma enkripsi *One-Time Pad* (*OTP*), tetapi tidak mengikuti *OTP* seutuhnya. Algoritma ini memanfaatkan suatu konsep unik dari ranah *Artificial Intelligent (*Kecerdasan Buatan*)-Artificial Immune System* (Sistem Imun Buatan) yaitu *Somatic Hypermutation*.

Algoritma *OTP* adalah pengenkripsi dan dekripsi yang tidak dapat dipecahkan jika mengikuti peraturan-peraturan yang ditentukan, jika mengambil struktur kunci yang keliru dapat menimbulkan suatu arti atau makna lain dari suatu *Ciphertext* yang didekripsinya (Soleh, M. & Hamokwarong, J. V., 2011). Ditambah lagi dengan pola kunci atau struktur kunci yang pernah digunakan sebelumnya tidak akan digunakan lagi, hal ini dilakukan agar meminimalisir terjadinya pembobolan akibat kesamaan pola atau struktur kunci (Ramadayanti, A. L. 2008). Sedangkan *Somatic Hypermutation* adalah suatu konsep dari sistem immunologi manusia yang secara sepenuhnya mengontrol mutasi suatu sel tertentu guna beradaptasi dengan antigen tertentu juga (Teng, G. & Papavasiliou, F. N., 2007. Dan Timmis, J., et al, 2008).

Somatic Hypermutation akan diterapkan ke dalam algoritma *OTP* dengan membuat suatu langkah baru di dalamnya. Penulis mengimplementasikan konsep *SHM* ke dalam *OTP* dengan mengharapkan tingkat keamanan algoritma *OTP* lebih tinggi lagi. Penulis memberatkan titik fokus pada perancangan algoritma baru yang menjadi masalah utama, dan juga memfokuskan satu tujuan spesifik yaitu mengemukakan algoritma yang ingin dibuat dengan memanfaatkan konsep *Somatic Hypermutation* ke dalam algoritma *One-Time Pad*.

2. LANDASAN TEORI DAN TINJAUAN PUSTAKA

2.1. Landasan Teori

2.1.1. *Vernam Cipher* (*One-Time Pad*)

Vernam Cipher merupakan algoritma kriptografi yang ditemukan oleh Mayor J. Maugborne dan G. Vernam (Rachmawanto, 2010). Algoritma ini merupakan algoritma berjenis *symmetric key* yang artinya bahwa kunci yang digunakan untuk melakukan enkripsi dan dekripsi merupakan kunci yang sama. Dalam proses enkripsi, algoritma ini menggunakan cara stream *cipher* di mana *cipher* berasal dari hasil *XOR* antara bit *Plaintext* dan bit *key*. Algoritma *Vernam Cipher* diadopsi dari *one-time pad cipher*, di mana dalam hal ini karakter diganti dengan bit (0 atau 1). (Rachmawanto, 2010)

Dalam proses enkripsi, *Ciphertext* diperoleh dengan melakukan penjumlahan modulo 2 satu bit *Plaintext* dengan satu bit kunci, seperti terlihat pada rumus di bawah ini:

$$C = (P + K) mod\ 2 \dots\dots\dots (1)$$

Di mana :

C = *Ciphertext* (pesan yang sudah dienkripsi/disandikan)

P = *Plaintext* (pesan yang ingin dienkripsi/disandikan)

K = Kunci

Sedangkan dalam proses dekripsi, untuk mendapatkan kembali *Plaintext*, diperoleh dengan melakukan penjumlahan *modulo* 2 satu bit *Ciphertext* dengan satu bit kunci :

$$P = (C - K) mod\ 2 \dots\dots\dots (2)$$

Pada *cipher*, bit hanya mempunya dua buah nilai, sehingga proses enkripsi hanya menyebabkan dua keadaan pada bit tersebut, yaitu berubah atau tidak berubah. Dua keadaan tersebut ditentukan oleh kunci enkripsi yang disebut dengan aliran-bit-kunci (*keystream* atau Kunci). Oleh karena operasi penjumlahan modulo 2 identik dengan operasi bit dengan operator *XOR*, maka persamaan dapat ditulis secara sederhana sebagai berikut:

$$C = (P)\ XOR\ (K) \dots\dots\dots (3)$$

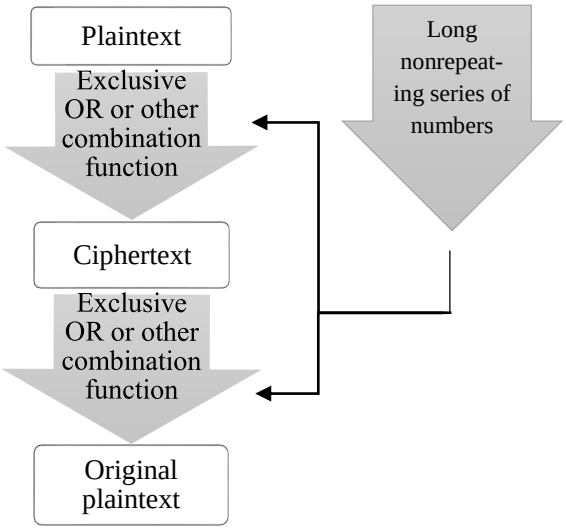
Sedangkan pada proses pendekripsian dituliskan:

$$P = (C)\ XOR\ (K) \dots\dots\dots (4)$$

Dalam operator logika *XOR*, hasil akan T (benar) apabila salah satu dari kedua *operand* (tetapi tidak keduanya) bernilai T atau 1. Atau dengan kata lain, apabila diaplikasikan dalam bit maka operator *XOR* akan menghasilkan 1 jika dan hanya jika salah satu *operand* bernilai 1. Sedangkan suatu bilangan dalam biner apabila di-*XOR*-kan dengan dirinya sendiri akan menghasilkan 0.

2.1.1.1. Proses Enkripsi *OTP*

Dalam algoritma *One-Time Pad* yang digambarkan pada Gambar 1, terdapat beberapa langkah untuk proses enkripsi dan dekripsi. Pertama-tama, karakter-karakter yang terdapat pada *Plaintext* dan kunci merupakan karakter *ASCII* (Rachmawanto, 2010). Maka, ubah *Plaintext* dan kunci menjadi bilangan biner. Lalu, kedua bilangan biner itu di-*XOR*-kan dan akan menghasilkan *Ciphertext* yang sudah



terenkripsi. (Rachmawanto, 2010)

Gambar 1. Algoritma Vernam Cipher (OTP) (Pfleeger, P. C., et al, 2015)

2.1.1.2. Proses Dekripsi *OTP*

Proses dekripsi dalam algoritma *Vernam Cipher* merupakan kebalikan dari proses enkripsi. *Ciphertext* dari hasil enkripsi di-*XOR*-kan dengan kunci yang sama (Gambar 1). (Rachmawanto, 2010)

2.1.2. *Somatic Hypermutation*

Somatic Hypermutation adalah salah satu konsep yang ditawarkan oleh proses imun tubuh dan bagaimana cara tubuh merespon suatu ancaman dari luar tubuh, *Artificial Immune System* atau *AIS*, yang adalah salah satu dari cabang *Artificial*

Intelligent. *Somatic Hypermutation* secara alami memiliki tujuan khusus yaitu memutasi suatu sel di tingkat kromosom secara terkendali untuk menghasilkan suatu Sel B yang lebih reaktif terhadap *antigen* (Al-Otaibi, S. T., & Ykhlef, M., 2015; Teng, G. & Papavasiliou, F. N., 2007; Timmis, J., et al, 2008).

2.2. Tinjauan Pustaka

Rachmawanto, E. H. (2010) membahas tentang pembuatan aplikasi steganografi yang memanfaatkan algoritma *Vernam Cipher (OTP)* sebagai pengenkripsi *file* yang akan dikirim nantinya. Aplikasi yang dibuat dapat menyelipkan suatu *file* utama ke dalam *file* lain dan menempatkannya di akhir *file* seperti bagaimana metode *End of File* seharusnya diimplementasikan sebelum *file* dienkripsi dan disimpan. Metodologi yang digunakan di dalam penelitian Rachmawanto adalah *RAD (Rapid Application Development)*. Peneliti memanfaatkan *Visual Basic* 6.0 sebagai media pembuatan aplikasinya.

Sugianto dan Yuniarto, T (2014) menggunakan algoritma *Mono Alphabetic* dengan algoritma *One-Time Pad* dengan tujuan untuk membuat algoritma dengan keamanan yang lebih baik. Sugianto dan Yuniarto juga membuat aplikasi pengenkripsiannya menggunakan *Visual Basic.*

Al-Otaibi, S. T. dan Ykhlef, M. (2015) membuat jurnal yang memanfaatkan *Artificial Immune System* untuk membuat suatu sistem yang dapat merekomendasikan seorang pelamar kerja berdasarkan persyaratan kerja yang dia penuhi dari banyak pekerjaan yang tersedia secara online.

Bisa dilihat dari Rachmawanto (2010) dan Sugianto (2014), mereka sama-sama menggunakan algoritma *OTP* sebagai dasar pengembangan algoritma yang mereka buat. Begitu juga penelitian ini, penulis juga mengembangkan algoritma *OTP* sebagai dasar algoritma yang dibuatnya.

Perbedaan antara Rachmawanto (2010), Sugianto (2014), dan penulis adalah konsep atau algoritma yang digunakan dalam pengembangan algoritma *OTP* saling berbeda. Rachmawanto menggunakan suatu konsep yang bernama steganografi, sedangkan Sugianto menggunakan algoritma *Mono Alphabetic* dalam pengembangannya. Lalu, Rachmawanto dan Sugianto sama-sama mengimplementasikan algoritma buatan mereka ke dalam bentuk aplikasi, sedangkan penulis hanya menggagaskan suatu algoritma baru yang menggunakan konsep *Somatic Hypermutation* ke dalam algoritma *OTP*.

Di samping itu, Rachmawanto dan Sugianto juga tidak menyentuh konsep-konsep yang disediakan oleh ranah *Artificial Inteligence*. Penulis memang tidak menggunakan konsep dari *Artificial Inteligence* secara eksplisit dan dapat dilihat langsung di mana letak kecerdasan buatan yang diimplementasikan, tetapi penulis menggunakan satu konsep minor yang berada di dalam subkategori *Artificial Immune System*, yaitu *Somatic Hypermutation*.

Al-Otaibi (2015) menggunakan dua tahap dalam *Artificial Immune System*, yaitu tahap *Clonning* dan *Mutation*, sedangkan penulis hanya menggunakan tahap *Mutation* dalam algoritma yang dibuatnya. Al-Otaibi tidak berkonsentrasi pada bidan keamanan seperti Rachmawanto, Sugianto, dan penulis, justru Al-Otaibi benar-benar fokus kepada ranah *Artificial Intelligent*, sehingga tidak begitu banyak kesamaan yang dapat diperhatikan pada penelitian Al-Otaibi dan penelitian ini.

3. HASIL DAN PEMBAHASAN

Algoritma baru yang dibuat akan tetap mengikuti peraturan-peraturan dari One-Time Pad yang ada, yaitu:

- Kunci diambil secara acak.
- Kunci yang sudah digunakan tidak akan digunakan lagi.

Dengan operasi matematis dari persamaan 3 dan 4 (Rachmawanto, 2010):

Didapatkan rumus:

$$C = E(P) \dots\dots\dots (5)$$

$$P = D(C) \dots\dots\dots (6)$$

Di mana:

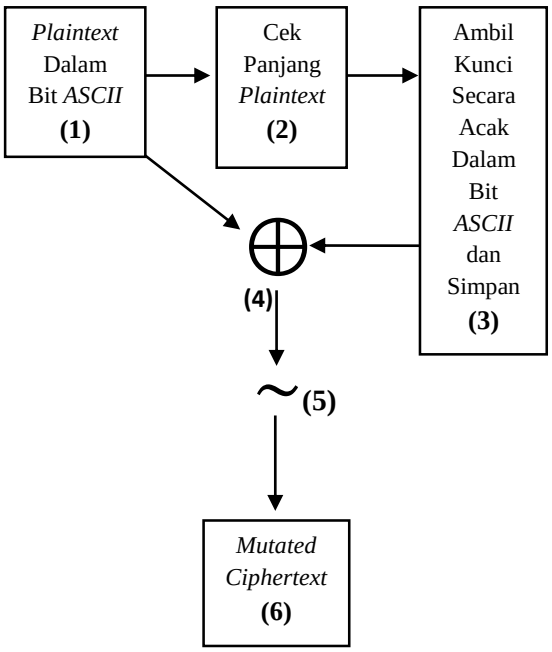
C = *Ciphertext*
P = *Plaintext*
E = Proses enkripsi
D = Proses dekripsi
K = Kunci
XOR = Operator *Exclusive OR*

3.1. Proses Enkripsi

Algoritma yang dibuat pada dasarnya masih menggunakan langkah-langkah dasar dari algoritma *OTP*. Dengan pengimplementasian konsep *SHM* ini, ditambahkan satu proses baru sebelum akhirnya proses enkripsi ini selesai. Langkah-langkah proses enkripsi terperincinya sebagai berikut:

Pada Gambar 2, proses yang dilakukan oleh komputer pertama-tama adalah langkah 1, yaitu *Plaintext* yang akan dienkripsi harus diubah menjadi deretan bit *ASCII*-nya sebelum melakukan proses yang lain. Proses ini berguna sebagai tahapan awal dari proses enkripsi di samping proses pengambilan kunci berjalan setelah panjang *Plaintext* telah ditentukan.

Di langkah ke-2, sebelum Kunci ditentukan, harus dilakukan pengecekan untuk mengetahui panjang *Plaintext* yang ingin dienkripsi. Seperti yang dicantumkan pada Gambar 2, langkah ke-3 dilakukan di mana Kunci diharuskan diambil dari deretan acak guna tidak adanya pola yang dapat menjadi cela pembobolan nantinya. Tahap ke-3 juga memastikan kunci harus dalam bentuk bit *ASCII* dan disimpan sebelum memasuki proses berikutnya.



Gambar 2. Proses Enkripsi

Setelah itu langkah ke-4, dilakukan proses operator *XOR* (*Exclusive OR*) antara *Plaintext* dengan Kunci yang sudah dalam bentuk bit *ASCII*-nya. Langkah ini adalah langkah terpenting dari algoritma *OTP* yang menghasilkan *Ciphertext*, tetapi *Ciphertext* yang dihasilkan oleh langkah ini masih belum bisa dianggap sebagai output dari algoritma yang dirancang.

Tahap ke-5 adalah tahap di mana pengimplementasian *Somatic Hypermutation* (*SHM*) berada. Seperti tujuan dasar dari *SHM*, proses ini ada untuk memroduksi suatu pertahanan yang lebih efektif melawan *antigen*, atau dalam kasus ini adalah hasil proses enkripsi yang menjadi lapisan awal pertahanan *Ciphertext* sebelum *Ciphertext* dapat diakses lebih

lanjut nantinya. Proses ini bertujuan untuk mengubah *Ciphertext* menjadi bentuk *Ciphertext* baru yang lebih membingungkan jika pembobol mencoba menyadapnya.

Proses ke-5 pada Gambar 2 ini menggunakan konsep *SHM* yang unik, yaitu mutasi secara terkendali (*Full-controled*). Peneliti merancang proses ini untuk memutarbalikkan semua nilai bit *ASCII* menggunakan operator *Not* (~) pada *Ciphertext* sebelum *Ciphertext* dihapus dan menghasilkan bentuk baru dari *Ciphertext* yang penulis sebut sebagai *Mutated Ciphertext* atau *Ciphertext* Yang Dimutasi (Gambar 2, nomor 6).

Mutated Ciphertext adalah hasil dari model algoritma baru yang dirancang. *Mutated Ciphertext* ini awalnya masih dalam bentuk bit *ASCII* dan akan secara otomatis menjadi karakter-karakter yang diacu oleh bit *ASCII*-nya.

Proses enkripsi yang dirancang dapat menjadi persamaan matematika sebagai berikut:

$$MC = \sim(C) \dots\dots\dots (7)$$

Jika persamaan 3 disubtitusikan ke dalam persamaan 7, maka akan menghasilkan:

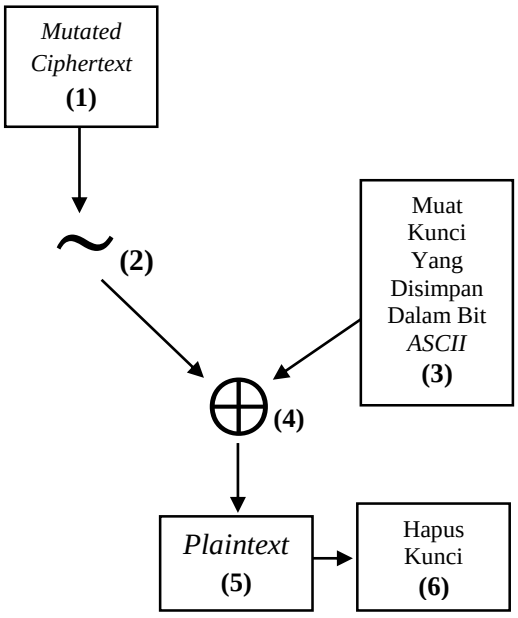
$$MC = \sim\big((P) XOR\ (K)\big) \dots\dots\dots (8)$$

Di mana:

MC = *Mutated Ciphertext*
C = *Ciphertext*
P = *Plaintext*
K = Kunci
XOR = Operator *Exclusive OR*

3.2. Proses Dekripsi

Proses dekripsi adalah kebalikan dari proses enkripsi. Proses ini diawali dengan *Mutated Ciphertext* (1) menjadi input sebelum memasuki langkah operasi *Not* seperti pada langkah 2 di Gambar 3. Langkah ini memutarbalikkan nilai bit *ASCII* dari *input* dan memiliki *Ciphertext* sebagai *output* dari langkah ini.



Gambar 3. Proses Dekripsi

Langkah ke-3, Kunci yang disimpan dari proses enkripsi pada langkah 3 di Gambar 2 akan dipanggil dalam bentuk bit *ASCII*-nya sebelum Kunci dan *Ciphertext* diproses oleh operator *XOR* (langkah 4, Gambar 3).

Setelah proses ke-4 di mana *XOR* antara *Ciphertext* dan Kunci dilakukan, hasil yang keluar adalah *Plaintext* (5) atau pesan yang dienkripsi tadi. Setelah *Plaintext* sudah didapatkan, algoritma ini akan memastikan bahwa Kunci yang sudah digunakan akan dibuang dan tidak akan dipakai lagi pada langkah terakhir dari Gambar 3, yaitu langkah 6.

Proses dekripsi ini dapat disimpulkan dalam bentuk matematis sebagai berikut:

$$P = \sim\big(D(C)\big) \dots\dots\dots (9)$$

Untuk membuktikan bahwa persamaan 7 dapat didekripsikan kembali ke P awal

maka dilakukan pembuktian matematis. Pada dasarnya dari persamaan 6 dilakukan perhitungan rumus di mana langkah-langkahnya sebagai berikut:

$$C = \sim(MC) \dots\dots\dots (10)$$
$$MC = \sim\big((P)\ XOR\ (K)\big) \dots\dots\dots (11)$$
$$C = \sim(\sim\big((P)XOR\ (K)\big) \dots\dots\dots (12)$$
$$C = (P)XOR\ (K) \dots\dots\dots (13)$$

Dengan dasar proses enkripsi dari persamaan 5, jika proses E dan D disamakan objek operasinya, maka:

$$E(x) = (x)XOR\ (K) \dots\dots\dots (14)$$
$$D(x) = (x)XOR\ (K) \dots\dots\dots (15)$$
$$D(x) = E(x) \dots\dots\dots (16)$$

Sehingga didapatkan:

$$D(x) = (x)XOR\ (K) \dots\dots\dots (17)$$

Subtitusi C:

$$D(C) = (C)XOR\ (K) \dots\dots\dots (18)$$

Sehingga menghasilkan:

$$P = (C)XOR\ (K) \dots\dots\dots (19)$$

Bisa dilihat bahwa persamaan 19 sama dengan persamaan 4.

Di mana:

MC = *Mutated Ciphertext*
C = *Ciphertext*
E = Proses enkripsi
D = Proses dekripsi
P = *Plaintext*
K = Kunci
XOR = Operator *Exclusive OR*

4. KESIMPULAN & SARAN

Penulis dapat mengambil kesimpulan bahwa prinsip algoritma yang dibuat menggunakan konsep *SHM* ke dalam algoritma *OTP* adalah penambahan satu langkah yang dianggap dapat meningkatkan keamanan dari algoritma *OTP*. Selain itu, algoritma yang dirancang juga cukup sederhana dan jika digunakan dengan benar sesuai aturan dan langkah-langkahnya maka pesan yang dienkripsi bisa dinyatakan aman.

Algoritma ini memiliki kelemahan pada penyimpanan kuncinya, di mana kunci yang disimpan dapat dibilang terancam akan penyadapan. Kunci tersebut adalah ketergantungan dari proses dekripsi, di mana proses tersebut tidak dapat dijalankan atau menghasilkan pesan yang salah jika kunci hilang atau kunci diubah. Penyimpanan dan authorisasi kunci juga bisa menjadi masalah.

Kunci satu bisa sama panjang dengan kunci yang lain dan ketika kunci dipanggil untuk dilakukan proses dekripsi, kunci bisa tertukar dengan kunci lain jika tidak adanya authorisasi. Selain itu, penulis juga menyadari kekurangan dari sisi penyimpanan kuncinya, di mana butuh penyimpanan yang besar untuk menyimpan kunci yang besar juga. Tetapi, penulis dapat menyimpulkan bahwa kelemahan-kelemahan algoritma yang dibuat kurang lebih sama dengan kekurangan milik algoritma *OTP* pada dasarnya.

Setelah dilakukan penelitian ini, penulis menyarankan penelitan-penelitian berikutnya dilakukan untuk menjawab beberapa bulir di bawah ini:

1). Algoritma yang dibuat belum pernah diuji dan diimplementasikan ke dalam program secara nyata. Oleh karena itu penulis menyarankan untuk pengimplementasian algoritma yang dirancang ke dalam program atau sistem keamanan seperti Rachmawanto, E. H. (2010), Sugianto dan Yuniarto, T. (2014) dan Soleh, M. dan Hamokwarong, J. V. (2011).

2). Penulis sejak awal hanya berasumsi bahwa keamanan dari algoritma yang dirancang melebihi

algoritma *OTP* pada dasarnya tetapi belum dilakukan uji coba dan pembuktian atas pernyataan tersebut, jadi sangat disarankan untuk melakukan perbandingan dan pembuktian akan keamanan algoritma yang sudah dirancang dengan algoritma *OTP* asli.

3). Penulis juga menyarankan untuk mengembangkan algoritma yang dibuat agar menjadi batu lompatan di dalam bidang kriptografi.

5. DAFTAR PUSTAKA

Al-Otaibi, S. T. & Ykhlef, M. 2015. "International Journal of Scientific Research and Innovative Technology". *Immunizing Job Recommender System*. 2(10), 97-110.

Pfleeger, P. C., Pfleeger, S. L, & Margulies, J. 2015. *Security In Computing Fifth Edition*. Prentice Hall.

Rachmawanto, E. H. 2010. *Teknik Keamanan Data Menggunakan Kriptografi Dengan Algoritma Vernam Cipher Dan Seganografi Dengan Metode End Of File (EOF)*. Universitas Dian Nuswantoro.

Ramadayanti A. L. 2008. *Analisa Algoritma Vernam (OTP).* Jurusan Teknik Informatika Universitas Sriwijaya.

Soleh, M. & Hamokwarong, J. V. 2011. "Momentum". *Aplikasi Kriptografi Dengan Metode Vernam Cipher Dan Metode Permutasi Biner*. 7(2), 8-13.

Sugianto & Yuniarto, T. 2014. "Jurnal Ilmiah SISFOTENIKA". *Kriptografi Gabungan Menggunakan Algoritma Mono Alphabetic Dan One-Time Pad*. 4(1), 53-63.

Supriyatna, A. 2014. "Prosiding Seminar Nasional Aplikasi Sains & Teknologi (SNAST)". *Analisis Tigkat Keamanan Sistem Informasi Akademik Dengan Mengkombinasikan Standar BS-7799 Dengan SSE-CMM*. A 181-188.

Teng, G. & Papavasiliou, F. N. 2007. *Immunoglobulin Somatic Hypermutation*. New York : Annu. Rev. Genet.

Timmis, J., Hone, A., Stibor, T. & Clark, E. 2008. "Theorical Computer Science" *Theorical Advances in Artificial Immune Systems*. 403(2008), 11-32.