Keamanan Pesan Teks Menggunakan Teori Chaos dan Electronic Code Book

Ervyn Yoga Indra[1], Muljono[2]
[1,2]Teknik Informatika, Universitas Dian Nuswantoro Semarang
Jl. Imam Bonjol 205-207 Semarang 50131
Email: [1]ervynskripsiku@gmail.com, [2]muljono@dsn.dinus.ac.id

Abstrak

Masalah keamanan merupakan salah satu aspek penting dalam system informasi. Dalam komunikasi pasti akan ada pengiriman pesan kepada orang lain, maka tentunya pesan tersebut harus sampai dengan aman. Sebuah informasi umumnya hanya ditunjukan bagi golongan tertentu, sangatlah pentinguntuk mencegah agar keamanan pesan tidak sampai jatuh kepada pihak-pihak lain yang tidak berkepentingan.Untuk mengurangi tindak kejahatan dalam keamanan data, maka kriptografi bias dijadikan solusi yang tepat. Dalam penelitian ini, teori Chaos dengan Logistic Map akan digunakan untuk membangkitkan kunci secara acak dan panjang. Kemudian kunci tersebut diterapkan pada algoritma Stream Cipher dan Electronic Code Book (ECB).Dengan teori Chaos tersebut akan dihasilkan kunci yang acak dan panjang kunci sama dengan panjang plainteks pada Stream Cipher. Sedangkan pada ECB akan menambah jumlah panjang kunci yang acak sehingga dapat menutup kelemahan. Dari hasil penelitian ini, teori Chaos dengan Logistic Map dapat mempermudah dalam mengingat kunci yang acak dan sekaligus panjang. Selain itu penelitian ini juga mengembangkan pada nomor iterasi tertentu yang dihasilkan oleh Logistic Map dapat dipilih menjadi nomor iterasi pertama kunci sehingga dalam pembangkitan kunci akan menambah variasi dan kemungkinan dalam penebakan kunci.

***Kata kunci** : kriptografi, chaos, logistic map, pembangkit kunci, stream cipher, electronic code book*

Abstract

Security problem is one of many important aspect in information system. In communications will be message sending to another people. So, definitly those message must arrive safely. An information generally should only be shown for certain people, it is very important to prevent message's security to fall to the wrong hands. To lessen criminal activity within data security, so cryptography can be use as the right solution. In this research, Chaos teory with logistic Map will be use to generate keys randomly and long. Then those keys implemented to Stream Cipher algorithm and Electronic Code Book (ECB). With these Chaos teory, random keys will be generated and key's length will be same with the plaintext in Stream Cipher. While in ECB, a random key's length will be added so weakness can be covered. From this research's result, Chaos teory with Logistic Map can make remembering random and long keys easier. Other than that, this research also developing in some certain iteration number which was generated by Logistic Map can be chosen as key's first iteration number, so in keys generation, variation and possibility in keys guessing will be increased.

***Keywords** : cryptography, chaos, logistic map, generate key, stream cipher, electronic code book*

1. Pendahuluan

Masalah keamanan merupakan salah satu aspek terpenting dalam sistem informasi. Namun hal ini sering kurang mendapat perhatian dari para perancang dan pengelola system informasi. Bahkan pada prioritasnya keamanan sering berada pada urutan setelah tampilan [1]. Dalam komunikasi pasti akan ada pengiriman pesan kepada orang lain, maka tentunya pesan tersebut harus sampai dengan aman. Terkadang pesan yang dikirim mengandung kerahasiaan.

Pada akhir-akhir ini telah terjadi kegiatan kejahatan yang dilakukan oleh negara-negara lain salah satunya yaitu penyadapan, sehingga merugikan dan mengganggu kedaulatan suatu negara yang menjadi korban termasuk Indonesia. Hal itu pastinya akan menimbulkan masalah yang serius dan apabila tidak cepat ditangani akan berakibat sangat buruk. Maka dari itu aspek keamanan sangat diperlukan untuk dijadikan pilihan utama.

Untuk mengurangi tindak kejahatan tentang keamanan data, maka kriptografi bisa dijadikan solusi yang tepat.Dalam kriptografi, banyak teknik yang dapat digunakan untuk melindungi pesan atau data dalam bentuk tulisan, suara, gambar bahkan video. Kriptografi dapat juga digabungkan dengan cabang ilmu lainnya sepert imatematika yang dapat dihubungkan dengan teori Chaos. Rinaldi Munir mengatakan bahwa sistem dinamis Chaos mempunyai property berharga di dalam kriptografi, yaitu peka pada perubahan kecil pada kondisi awal sistem. Sifat peka ini berarti bahwa dua nilai awal dipilih sangat dekat satu sama lain, maka setelah sejumlah iterasi tertentu barisan nilai yang dihasilkan akan berbeda secara signifikan [2]. Dengan menggunakan Logistic Map, dapat menghasilkan kekacauan pada masing-masing urutan sirklus. Sehingga hasil urutan sebelumnya dapat mempengaruhi hasil cipherteks selanjutnya [3].

Logistic Map dapat diterapkan pada kriptografi karena sifatnya yang kacau.Karenasifatnya yang kacau itulah sangat cocok untuk memenuhi syarat kriptografi yang kuat.Logistic Map yang merupakansalahsatu Chaotic Map mempunyai kekurangan bila diterapkan pada kriptografi. Kekurangannya adalah pada dua nilai awal yang dihasilkan dapat diprediksi [4]. Dari penelitian tersebut penulis mengembangkan dengan menetapkan dalam pemilihan nomor iterasi tertentu sebagai deret awal kunci. Keamanan pesan dalam bentuk tulisan sangatlah penting karena banyak sekali digunakan untuk pertukaran informasi. Dan teknik yang digunakan yaitu Stream Cipher dan Electronic Code Book (ECB) denganteori Chaos sebagai pembangkit kunci.Untuk cipherteks yang dihasilkan adalah berupa bilangan hexadecimal yang lebih cocok diterapkan dari pada bilangan yang lainnya sehingga dari penelitian ini aspek keamanan pada kerahasiaan pesan dapat terpenuhi.

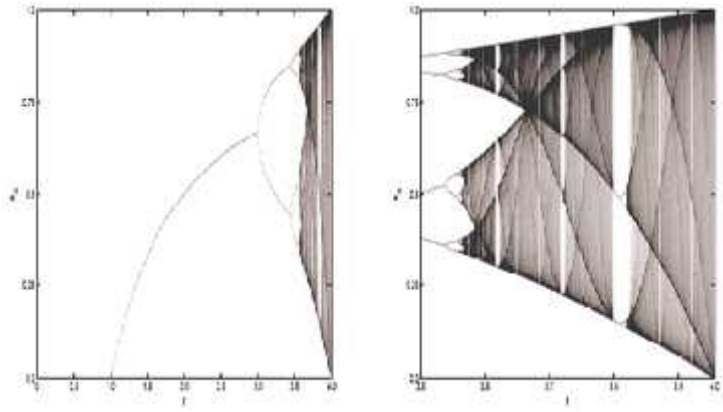
2. Metode Penelitian

2.1. Teori Chaos

Teori chaos adalah teori yang menggambarkan perilaku sistem dinamis nirlanjar yang menunjukan fenomena yang kacau. Salah satu teori sistem chaos adalah sangat peka terhadap nilai awal. Hal ini menunjukan hasil yang sangat kacau jika nilai awal berbeda sedikit saja. Map dari suatu nilai tertentu yang polanya sangat sensitif terhadap perubahan disebut *Chaotic Map*. Ada banyak chaotic map yang telah ditemukan, salah satunya adalah *Logistic Map*. *Logistic Map* merupakan salah satu fungsi chaos sederhana di dalam ekologi yang digunakan untuk mensimulasikan pertumbuhan populasi spesies. *Logistic Map* juga merupakan satu dimensi yang telah digunakan secara luas, yang didefinisikan sebagai berikut :

$$X_{i+1} = r\ X_i (1 - X_i) \qquad (1)$$

Dari persamaan di atas, X adalah populasi spesies pada interval waktu yang ditentukan dengan X_0 adalah nilai awal iterasi. Daerah asal X adalah dari 0 sampai 1, yang dalam hal ini 1 menyatakan populasi maksimum dan yang 0 menyatakan kepunahan, sedangkan $0 \leq r \leq 4$. Konstanta *r* menyatakan laju pertumbuhan. Konstanta *r* juga menyatakan bagian nirjalar dari persamaan. Ketika *r* meningkat, maka sistem juga naik.



Gambar 1. Diagram *bifurcation* untuk persamaan logistik

Gambar 1. memperlihatkan kelakuan fungsi yang dalam hal ini sumbu-*x* menyatakan nilai *r* sedangkan sumbu-*y* menyatakan status sistem, yaitu nilai x. Bila $0 < r < 1$, nilai awal berapapun akan menghasilkan kepunahan. Bila $1 < r < 3$, fungsi konvergen ke sebuah nilai *(fixed-point)*, yaitu nilai *r* yang menghasilkan sistem yang mempunyai periode satu siklus. Ketika $r = 3$, kurva fungsi terpecah menjadi

dua *(bifurcation)* menghasilkan dua nilai populasi yang berbeda, yang berarti nilai X secara periodik berosilasi dari status tinggi ke status rendah. Periode sistem pada nilai *r* ini adalah dua. Ketika *r* meningkat lagi, kurva fungsi terpecah lagi menjadi empat, yang berarti nilai-niklai X yang dihasilkan berosilasi di antara 4 nilai. Periode sistem pada nilai *r* ini adalah empat [2] [2].

Demikian seterusnya *bifurcation* menjadi lebih cepat lagi dengan meningkatnya nilai *r* sampai tiba pada suatu nilai *r* tertentu sifat chaos pun muncul. Pada titik ini tidak mungkin lagi memprediksi kelakuan sistem. Kita dapat melihat bahwa ketika *r*> 3.75 sistem mulai melaju dengan cepat menuju area chaos (gambar yang diasir). Akhirnya, ketika *r* = 4, iterasi bergantung sepenuhnya terhadap nilai awal atau X_0 dan nilai-nilai yang dihasilkan muncul acak meskipun sistem ini deterministik. Nilai-nilai chaos yang dihasilkan akan berada di dalam rentang yang lengkap anatara 0 dan 1 [5] [5].

Fungsi *Chaos* yaitu menggunakan *Logistic Map* banyak digunakan pada kriptografi. Karena *Logistic Map* mempunyai kesensitifan pada nilai awal sehingga menghasilkan kekacauan. Kekacauan tersebut diterapkan pada kunci pada kriptografi.Sehingga cocok untuk dikembangkan di masa depan [6] [6].

2.2. *Stream Cipher*

Aliran kode *(cihper stream)* mengenkripsi teks-asli menjadi teks-kode bit per bit (1 bit setiap kali transformasi). Pertama kali diperkenalkan oleh Vernam melalui algoritma yang dikenal dengan nama kode Vernam. Cipher aliran merupakan versi lain dari *one-time-pad*.

Satu-satunya algoritma kriptografi yang sempurna aman dan tidak dapat dipecahkan adalah one the pad (secara matematis Shannon telah membuktikan bahwa OTP tidak dapat dipecahkan). OTP ditemukan pada tahun 1917 oleh Vernam dan Major Joseph Mauborge. *One Time Pad* (*pad* = kertas bloknot) adalah kertas yang berisi deretan karakter-karakter kunci yang berisi huruf-huruf yang tersusun acak. Enkripsi dapat dinyatakan sebagai penjumlahan modulo 26 dari satu karakter plainteks dengan satu karakter kunci one-time pad :

$$C_i = (P_i + K_i) \bmod 26 \qquad (2)$$

Setelah pengirim mengenkripsikan pesan dengan kata kunci, ia menghancurkan kunci tersebut (oleh karena itu disebut sekali pakai atau *one-time*). Penerima pesan menggunakan kunci yang sama untuk mengdekripsikan karakter-karakter cipherteks menjadi karakter-karakter plainteks dengan persamaan:

$$P_i = (C_i - K_i) \bmod 26 \qquad (3)$$

Shanon membuktikan apabila sandi *one time pad* diterapkan secara benar maka akan mencapai rahasia sempurna, (Shannon, 1949). Sebuah sandi disebutkan demikian bila pasangan teks asli dan teks sandi tidak memiliki hubungan statistik sehingga sulit bagi penyerang untuk melakukan analisis sandi atau analisis statistik [7].

Stream Cipher yang terdahulu secara umum dapat diserang, akan tetapi untuk desain *Stream Cipher* yang baru sangatlah sulit. Perlu banyak tambahan teknik dan literatur yang perlu dikembangkan untuk kriptanalisis. Sehingga perlu adanya pembaharuan untuk menyerang sesuatu yang juga baru didesain [8].

2.3. *Electronic Code Book (ECB)*

Blok kode merupakan suatu algoritma yang masukan dan keluarannya berupa satu blok, dan setiap bloknya terdiri dari beberapa bit (1 blok terdiri dari 64 bit, 128 bit, atau adakalanya lebih).

Electronic Code Book (ECB) adalah salah satu teknik yang digunakan pada Cipher Blok. Pada mode ini, setiap blok palainteks P_i dienkripsi secara individual dan independen menjadi blok cipherteks C_i. Mode ini tidak akan mempengaruhi blok-blok lainnya. Istilah "*code book*" di dalam ECB muncul dari fakta bahwa karena blok plainteks yang sama selalu dienkripsi menjadi blok cipherteks yang sama, maka secara teoritis dimunkinkan membuat buku kode plainteks dan cipherteks yang berkoresponden. Namun, semakin besar ukuran blok, semakin besar ukuran buku kodenya. Misalkan jika blok berukuran 64 bit, maka buku kode terdiri dari $2^{64} - 1$ buah kode *(entry)*, yang berarti terlalu besar untuk disimpan. Lagipula, setiap kunci mempunyai buku kode yang berbeda. Secara matematis, enkripsi dengan mode ECB dinyatakan sebagai :

$$C_i = E_k(P_i) \qquad (4)$$

dan dekripsi sebagai :

$$P_i = D_k(C_i) \qquad (5)$$

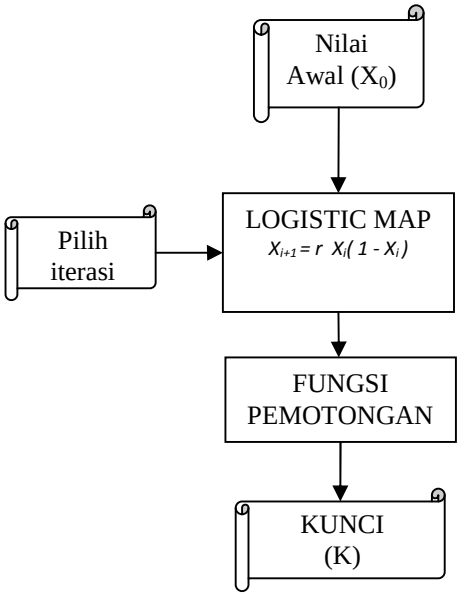
Ada kemungkinan panjang plainteks tidak habis dibagi dengan panjang ukuran blok yang ditetapkan (misalkan 64 bit atau yang lainnya). Hal ini mengakibatkan blok terakhir berukuran lebih pendek daripada blok-blok lainnya. Satu cara untuk mengatasi hal ini adalah dengan padding, yaitu menambahkan blok terakhir dengan pola bit yang teratur agar panjangnya sama dengan blok yang ditetapakan. Misalnya ditambahkan bit 0 semua, atau bit 1 semua, atau bit 0 dan bit 1 berselang seling. Misal ukuran blok adalah 64 bit (8 byte) dan blok terakhir terdiri dari 24 bit (3 byte). Tambahkan blok terakhir dengan 40 bit (5 byte) agar menjadi 64 bit, misalnya dengan menambahkan 4 buah byte 0 dan satu buah byte angka 1. Setelah dekripsi, hapus 5 byte terakhir dari k blok dekripsi terakhir. *Electronic Code Book* (ECB) bisa dijadikan sebagai awal proses sebenarnya sebelum proses enkripsi. Sehingga akan memperkuat teknik algoritma lain yang diterapkan [9].

3. HASIL DAN PEMBAHASAN

Kebutuhan sistem pada penelitian ini adalah pembangkit kunci, penerapan pembangkit kunci pada enkripsi dan dekripsi masing-masing algoritma *Stream Cipher* pada *One Time Pad* dan *Electronic Code Book* (ECB). Pada pembangkit kunci, penulis akan menjelaskan cara menerapkan teori *Chaos* dengan rumus *Logistic Map* untuk membuat deretan kunci sampai panjang yang akan ditentukan dan menghasilkan deretan nilai secara acak. Setelah deretan kunci dibangkitkan, deretan kunci tersebut akan digunakan untuk menjalankan penyandian (enkripsi) untuk menghasilkan kode hexadesimal. Dan deretan kunci yang sama juga digunakan untuk menjalankan pengembalian (dekripsi) sehingga akan mengembalikan teks asli yang sebelumnya sudah disandikan. Pada kedua algoritma tersebut, masing-masing berbeda dalam cara menyandikan (enkripsi) dan mengembalikan (dekripsi).

3.1. Pembangkitan Kunci Chaos

Untuk membangkitkan kunci yang acak, penulis menggunakan teori chaos yang mempunyai properti yang berharga bagi kriptografi. Dalam teori chaos terdapat rumus *Logistic Map* sebagai pembangkit kunci acak yang akan menghasilkan iterasi berdasarkan nilai awal. Di bawah ini adalah desain untuk pembangkit kunci dalam penelitian ini :



Gambar 2. Proses urutan pembangkit kunci

Pembangkit kunci dengan *Logistic Map* yaitu tergantung pada nilai awal (X_0) yang dimasukan. Dengan menerapkan rumus :

$$X_{i+1} = r\ X_i(1 - X_i) \qquad (1)$$

X_i : iterasi ke-i
r : konstanta

Konstanta *r* adalah 4 untuk memenuhi ketergantungan sistem terhadap nilai awal (X_0) yang sebelumnya telah dimasukan. Pada penelitian ini, dapat pemilihan nomor iterasi tertentu dapat dijadikan deretan awal dalam pembangkitan kunci. Lalu iterasi pertama (X_1) dihasilkan dan berhenti sebanyak kunci yang akan dibangkitkan (Xn). Sehingga akan dihasilkan deretan nilai *Chaos* yaitu X_1, X_2, X_3, X_4, X_5,...Xn.

Setelah deretan nilai *Chaos* dihasilkan, selanjutnya akan dilakukan proses pemotongan untuk mendapatkan nilai integer. Cara untuk mendapatkan nilai integer adalah dengan mengambil sebanyak tiga angka terbelakang.

Dari masing-msing deretan nilai Chaos yang dihasilkan dan sudah mengalami proses pemotongan untuk mendapatkan nilai integer yaitu tiga angka terbelakang maka kunci sudah terbentuk. Yaitu dari X_1, X_2, X_3, X_4, X_5,...Xn setelah mengalami proses pemotongan menjadi K_1, K_2, K_3, K_4, K_5,...Kn.

3.2. Penerapan kunci Chaos pada Stream Cipher

Stream Cipher mengenkripsi dengan menambahkan plainteks dengan kunci dan dilanjutkan dengan modulo 256. Perhitungan integer berdasarkan nomor ASCII.

Plainteks : { P_1, P_2, P_3, P_4, ..., Pn }
Kunci : { K_1, K_2, K_3, K_4, ..., Kn}
_____mod 256_____(+)
Cipherteks: { C_1, C_2, C_3, C_4, ..., Cn }

Setelah deretan desimal dari cipherteks terbentuk lalu diubah menjadi deretan hexadesimal. Sebelum melakukan dekripsi, pada cipherteks ditambah dengan 256 sejumlah kali sampai lebih besar dari elemen pad atau kunci, maka cipherteks dapat dikurangi oleh nilai dari deretan kunci yang sudah dibangkitkan sebelumnya.

Untuk menciptakan dari deretan cipherteks asli ke yang baru akan dilakukan proses dan kondisi sebagai berikut :

$$C_{nb} = (256 \text{ X } n) + C_{nl} \qquad (6)$$

lalu ambil nilai n terkecil yang memenuhi kondisi ($C_{nb} \geq K_n$).
n : konstanta
C_{nb} : Cipherteks baru
C_{nl} : Cipherteks lama

Sehingga dari deretan cipherteks lama {C_{nl1}, C_{nl2}, C_{nl3}, C_{nl4}, C_{nl5}, ..., C_{nl}n}
menjadi deretan cipherteks baru

{C_{nb1}, C_{nb2}, C_{nb3}, C_{nb4}, C_{nb5}, ..., C_{nb}n}.
Cipherteks:{C_{nb1},C_{nb2},C_{nb3},C_{nb4},...,C_{nb}n}
Kunci : { K_1, K_2, K_3, K_4, ..., Kn}
_____mod 256_____(+)
Plainteks : { P_1, P_2, P_3, P_4, ..., Pn }

Setelah deretan desimal dari plainteks terbentuk lalu diubah menjadi deretan plainteks string asli.

3.3. Penerapan kunci Chaos pada Electronic Code Book (ECB)

Pada mode ECB perlu memperhatikan terlebih dahulu pembagian blok yang akan ditentukan. Karena jumlah bit dalam blok harus sama dengan jumlah kunci yang dibangkitkan, maka iterasi pada pembangkit kunci berhenti sampai pada urutan bit terakhir blok yang sebelumnya sudah ditentukan. Misalkan tiap blok 40 bit atau berisi 5 byte (5 karakter) maka kunci yang dibangkitkan dengan *Chaos* juga sama yaitu 5 kali iterasi. Dan perlu dilakukan proses modulo 256 untuk merubah nilai integer supaya tidak melebihi 255 karena ASCII mempunyai maximal decimal tersebut agar dapat diubah menjadi biner.

Pembangkitan kunci biner :
Kunci :{ K_1, K_2, K_3, K_4 , ..., Kn}
_____ mod 256_____
Kunci biner:{ Kb_1,Kb_2,Kb_3,Kb_4,...,Kbn}

Plainteks : { P_1, P_2, P_3, P_4, P_5 ,..., Pn} dan dibagi menjadi sebanyak m-blok. Sehingga deretan biner dari tiap karakter tersebut menjadi terpecah-pecah. Dan deretan plainteks biner tersebut lalu dilakukan proses XOR dengan deretan kunci biner.

Blok ke-1 Blok ke-2 Blok ke-m
{P1,...,Pn} | {P1,...,Pn} | {P1,..,Pn}
{Kb1,..,Kbn}|{Kb1,..,Kbn}|{Kb1,..,Kbn}
⊕
Wrapping (pergeseran satu bit ke kiri)
{ C1, ..., Cn}|{ C1, ..., Cn}|{ C1, ..., Cn}

Sehingga cipherteks yang didapat merupakan hasil dari penggabungan tiap blok-blok yaitu dari blok pertama sampai dengan blok yang terakhir sesuai dengan urutan masing-masing blok. Lalu deretan cipherteks tersebut diubah menjadi deretan hexadesimal.

Di bawah ini adalah cara penerapan untuk menghasilkan deretan plainteks. Dari deretan cipherteks tersebut diubah menjadi deretan biner. Cara pertama adalah melakukan wrapping (pergeseran bit ke kanan) lalu deretan cipherteks biner mengalami proses XOR dengan deretan kunci biner.

Blok ke-1 Blok ke-2 Blok ke-m
{ C_1, ..., Cn}|{ C_1, ..., Cn}|{ C_1, ..., Cn}
Wrapping(pergeseran satu bit ke kanan)
{Kb_1,..,Kbn}|{Kb_1,..,Kbn}|{Kb_1,..,Kbn}
⊕
{P_1,...,Pn} | {P_1,...,Pn} | {P_1,..,Pn}

Sehingga plainteks yang didapat merupakan hasil dari penggabungan tiap blok-blok yaitu dari blok pertama sampai dengan blok yang terakhir sesuai dengan urutan masing-masing blok. Kemudian dari deretan plainteks biner tersebut dijadikan kembali menjadi karakter string asli.

4. KESIMPULAN

Teori Chaos Logistic Map dapat diterapkan pada algoritma kriptografi yaitu Stream Cipher dan Electronic Code Book (ECB). Teori Chaos Logistic Map dapat membangkitkan kunci acak dan mempermudah dalam mengingat kunci. Pada algoritma kriptografi kunci umumnya banyak karakter yang digunakan sehingga terkesan panjang dan kunci yang baik adalah kunci yang acak menambah kesulitan dalam mengingat kunci. Penerapan teori Chaos pada algoritma Stream Cipher lebih mendekati syarat keamanan yang sempurna karena panjang kunci sama dengan plainteks dari pada algoritma Electronic Code Book (ECB). Pada nilai awal pembangkitan kunci yaitu 0.25, 0.5 dan 0.75 akan menghasilkan iterasi yang nilainya sama sehingga keamanan menjadi percuma.

DAFTAR PUSTAKA

[1] D. Ariyus, Computer Security, Yogyakarta: Andi, 2007.
[2] R. Munir, Kriptografi, Bandung: Informatika Bandung, 2006.
[3] F. Wang, Y. Zhang dan T. Cao, “Research of Chaotic Block Cipher Algorithm Based on Logistic Map,” dalam Second International Conference on Intelligent Computation Technology and Automation, Xuzhou Jiangsu, 2009.
[4] M. Mishra dan H. V. Mankar, “Chaotic Encryption Scheme Using 1-D Chaotic Map,” Int. J.Communications, Network and System Sciences, vol. 4, no. 4, pp. 452-455, 2011.
[5] R. Munir, B. R. dan S. S. , “Perancangan Algoritma Kriptografi Stream Cipher dengan Chaos,” Institut Teknologi Bandung, Bandung, 2005.
[6] P. J. Rani dan S. D. Bhavani, “Symmetric Encryption Using Logistic Map,” IEEE, vol. 5, no. 12, 2012.
[7] R. Sadikin, Kriptografi untuk Keamanan Jaringan, Yogyakarta: Andi, 2012.
[8] U. M. Bokhari, S. Alam dan S. F. Masoodi, “Cryptanalysis Techniques for Stream Cipher : A Survey,” International Journal of Computer Application, vol. 60, no. 9, pp. 0975-8887, 2012.
[9] I. F. Elashry, O. S. Farag Allah dan A. M. Abbas, “A New Diffusion Mechanism for Data Encryption in The ECB Mode,” IEEE, vol. 8, no. 9, pp. 4244-5844, 2009.