Pendiskritan Pembangkit Bilangan Acak Peta Logistik Menggunakan Fungsi Trigonometri Osilasi Tinggi

Achmad Dimas Noorcahyo - 13508076
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
if18076@students.if.itb.ac.id

***Abstract*—Pembangkit bilangan acak memiliki peran penting dalam kriptografi kunci publik maupun kunci simetri. Dalam perkembangannya, teori *chaos* menjadi basis yang digunakan dalam metode komputasi pembangkit bilangan acak. Fungsi peta logistik sebagai salah satu fungsi pembangkit bilangan acak berbasiskan *chaos* dikenal sebagai fungsi pembangkit yang memiliki rumus penghitungan sederhana dan menghasilkan bilangan dengan tingkat keacakan tinggi. Meskipun demikian, pembangkit bilangan acak fungsi logistik memiliki kekurangan yaitu hanya dapat beroperasi dalam bilangan real. Untuk aplikasi kriptografi, pembangkit bilangan acak harus beroperasi dalam bilangan bulat berformat bit. Dalam makalah ini, ditawarkan metode pengembangan peta logistik yang bertujuan untuk mendiskritkan hasil bilangan acak peta logistik dari bilangan real menjadi bilangan bulat berbentuk bit satu atau nol. Metode ini juga dirancang untuk menambah tingkat keacakan dari pembangkit bilangan acak peta logistik. Prinsip mendasar dari metode yang ditawarkan ini adalah penggunaan fungsi trigonometri sinus dengan parameter $1/x^3$. Fungsi ini dipilih karena sifat osilasinya yang sangat acak di sekitar nilai nol. Batas nilai hasil bilangan acak dari pembangkit bilangan logistik dipersempit terlebih dahulu agar lebih dekat ke nilai nol lalu kemudian dijadikan parameter x untuk penghitungan fungsi sinus yang telah disiapkan. Jika hasil merupakan nilai positif maka dihasilkan bit 1 dan jika hasil merupakan nilai negatif maka dihasilkan bit 0. Analisis hasil dilakukan dengan cara melakukan pengujian keacakan statistik *Runs Test* pada deretan bit acak yang dihasilkan metode ini. Berdasarkan pengujian, dapat disimpulkan bahwa metode yang ditawarkan berhasil mengubah hasil bilangan acak peta logistik dari bilangan real menjadi deretan bit bulat dengan tingkat keacakan lebih baik.**

***Index Terms*— Fungsi trigonometri osilasi tinggi, Pembangkit bilangan acak, Pendiskritan, Peta logistik.**

I. PENDAHULUAN

Deretan bilangan acak merupakan deretan bilangan yang tidak dapat diprediksi kemunculannya. Deretan ini memiliki peran yang sangat penting dalam kriptografi kunci publik maupun kunci simetri. Dalam kriptografi kunci publik, bilangan acak digunakan untuk pembangkitan parameter kunci, sedangkan aplikasi bilangan acak juga penting untuk pembangkitan *Initialization Vector* (IV) pada algoritma kunci simetri. Karena perannya yang besar, maka dibutuhkan pembangkit bilangan acak yang aman. Pembangkit yang aman menghasilkan bilangan yang benar-benar acak, tidak dapat diprediksi, dan tidak berulang. Munculnya teori *chaos* menjawab kebutuhan ini. Teori *chaos* menggambarkan karakteristik sistem yang sangat peka terhadap nilai awal. Akibatnya kelakuan sistem yang memiliki sifat *chaos* akan muncul secara acak [1]. Terdapat banyak sistem *chaos* yang direpresentasikan oleh fungsi yang disebut peta. Salah satu peta yang banyak digunakan adalah peta logistik. Karena sifatnya yang *chaos*, peta logistik dapat digunakan sebagai pembangkit bilangan acak yang handal. Tingkat keacakan dari peta logistik sangat baik dan juga tidak tidak berulang karena tidak memiliki periode. Namun, peta logistik memiliki kekurangan. Peta logistik membangkitkan bilangan berbentuk real dari selang 0 sampai 1. Padahal, dalam aplikasinya pada kriptografi, bilangan acak yang dibutuhkan adalah bilangan berbentuk bulat, bahkan terkadang dalam bentuk bit saja. Untuk itu dibutuhkan suatu metode tersendiri untuk mendiskritkan nilai yang dibangkitkan oleh peta logistik. Pada makalah ini diusulkan sebuah metode pendiskrtitan bilangan acak yang berasal dari peta logistik. Metode yang akan diusulkan ini bertujuan bukan hanya untuk melakukan pendiskritan nilai bilangan acak, namun juga untuk meningkatkan tingkat keacakan dari pembangkit peta logistik. Demi mencapai tujuan tersebut, setelah melakukan penghitungan peta logistik, penulis kembali membangkitkan bilangan dengan sebuah fungsi yang juga bersifat acak pada selang tertentu yaitu fungsi $\sin(1/x^3)$. Kali ini nilai yang dibangkitkan berupa nilai bit 1 atau 0. Metode pendiskritan ini diharapkan dapat melengkapi peta logistik sebagai pembangkit bilangan acak yang handal, kali ini dengan deretan nilai keluaran yang berbentuk bit diskrit serta tingkat keacakan lebih tinggi.

II. PEMBANGKIT BILANGAN ACAK *CHAOS* : PETA LOGISTIK

A. Teori Chaos

Chaos adalah sebuah perilaku sistem deterministik yang memiliki karakteristik sangat peka terhadap perubahan nilai awal [6]. Karena sensifitasnya, kelakuan sistem yang bersifat *chaos* tampil secara acak [1].

Fungsi yang memiliki perilaku *chaos* sering direpresentasikan sebagai relasi rekurens yang disebut juga dengan peta [3]. Perilaku peta-peta *chaos* yang acak membuat prinsip *chaos* ini sangat berguna sebagai dasar komputasi pembangkit bilangan acak. Dalam perkembangannya, dikenal berbagai macam peta *chaos*. Salah satunya yang dikenal luas adalah peta logistik [2].

B. Peta Logistik

Fungsi logistik adalah salah satu fungsi yang memiliki karakteristik *chaos*. Fungsi logistik memiliki persamaan sebagai berikut [1] :

$$f(x) = rx(1-x) \quad (1)$$

Dalam komputasi, fungsi logistik biasanya dijalankan dalam bentuk persamaan iteratif yang memproduksi nilai baru di setiap lelarannya. Oleh karena itu, fungsi logistik memiliki bentuk persamaan iteratif yang disebut peta logistik. Peta logistik dinyatakan sebagai berikut [1] :

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

Dengan parameter-parameter :

r = laju pertumbuhan ($0 \leq r \leq 4$)
x= nilai *chaos* ($0 \leq x \leq 1$)

Persamaan peta logistik versi iteratif inilah yang digunakan sebagai pembangkit bilangan acak. Perilaku keacakan persamaan logistik terlihat pada grafik bifurcation berikut [6] :

Gambar 1. Grafik fungsi bifurcation dari peta logistik

Dari grafik terlihat bahwa nilai laju pertumbuhan r yang terletak di bagian kiri interval (dekat ke 0) tidak menghasilkan deretan bilangan yang acak. Semakin r mendekat ke nilai 4, maka hasil yang dibangkitkan peta logistik akan semakin menunjukkan perilaku random. Oleh karena itu, pada makalah ini nilai laju pertumbuhan selalu dipilih r = 4.

Untuk membangkitkan bilangan acak dengan peta logistik dibutuhkan sebuah nilai inisialisasi x_0 dalam interval [0,1]. Dengan satu nilai inisialisasi ini, persamaan peta logistik dapat mulai membangkitkan bilangan-bilangan acak secara iteratif. Sifat *chaos* menyebabkan deretan bilangan yang dihasilkan yaitu $x_1, x_2, x_3, \ldots$ menjadi sulit diprediksi.

Deretan bilangan yang dihasilkan peta logistik merupakan bilangan acak real yang berada pada interval [0,1]. Untuk mendapatkan bilangan acak bulat, dibutuhkan metode pendiskritan tersendiri, salah satunya yang diusulkan dalam makalah ini.

III. FUNGSI TRIGONOMETRI OSILASI TINGGI

Fungsi trigonometri memiliki sifat yang unik yaitu sifat grafiknya yang berosilasi sepanjang daerah definisi absisnya. Fungsi trigonometri sinus dan cosinus standar

logistik.

Jika pangkat dari parameter x ditingkatkan, perilaku grafik fungsi akan semakin rapat pada interval sekitar 0. Oleh karena itu, untuk mengoptimalkan tujuan yang ingin dicapai namun tetap menjaga kesederhanaan komputasi penulis menggunakan pangkat 3 untuk parameter x fungsi. Maka, fungsi yang digunakan seterusnya dalam makalah ini adalah :

$$y = \sin\left(\frac{1}{x^3}\right) \quad (4)$$

IV. METODE YANG DIUSULKAN

Metode yang diusulkan dalam makalah ini berdasar pada pemanfaatan fungsi sinus osilasi tinggi untuk menjadikan bilangan acak hasil pembangkitan peta logistik yang semula bilangan real menjadi bilangan bulat bit 1 atau 0. Metode ini bermula dari penghitungan peta logistik dengan parameter r = 4. Nilai inisialisasi dipilih sebarang dalam interval [0,1]. Peta logistik akan menghasilkan nilai acak pada setiap lelaran yang berbentuk bilangan real pada selang [0,1]. Bilangan real acak yang dihasilkan dibagi 10 terlebih dahulu untuk mempersempit selang kemungkinan nilai acak menjadi [0,0.1] sehingga lebih dekat ke nilai 0. Setelah itu nilai ini akan dijadikan parameter x penghitungan fungsi sinus berosilasi tinggi $y=\sin(1/x^3)$. Jika hasil yang didapat adalah bilangan positif atau nol, maka nilai bit 1 dibangkitkan, sedangkan jika hasil yang didapat adalah bilangan negaif, maka nilai bit 0 dibangkitkan. Proses ini dilakukan pada setiap lelaran. Hasil yang didapat berupa deretan bit acak. Secara runtut, metode pendiskritan hasil peta logistik menggunakan fungsi trigonometri osilasi tinggi dilakukan dalam langkah-langkah berikut :

1. Memilih nilai inisialisasi x_0 sebagai masukan peta logistik
2. Menghitung nilai acak x lelaran selanjutnya dengan persamaan :
   $$x_{i+1} = 4x_i(1 - x_i)$$
3. Membagi nilai acak x yang dihasilkan oleh persamaan pada langkah 2 dengan bilangan 10.
4. Menghitung nilai fungsi sinus osilasi tinggi :
   $$y = \sin\left(\frac{1}{x^3}\right)$$
   dengan parameter x adalah hasil penghitungan yang didapat pada langkah 2.
5. Jika nilai y yang didapat merupakan bilangan positif atau nol maka dibangkitkan bit 1, sedangkan jika nilai y yang didapat bilangan negatif maka dibangkitkan bit 0.
6. Ulangi langkah 2 hingga langkah 6 sesuai jumlah iterasi yang ditentukan
7. Didapatkan hasil berupa deretan bit acak.

Jika ditinjau lebih lanjut, metode ini merupakan pengembangan dari pembangkitan bilangan acak peta logistik. Hasil peta logistik yang didapat dipersempit intervalnya agar sedekat mungkin dengan 0. Hal ini perlu dilakukan sebelum hasil ini dilewatkan ke fungsi sinus osilasi tinggi $y=\sin(1/x^3)$ yang hanya berperilaku sulit diprediksi pada parameter x yang dekat ke 0. Jika x yang dihitung cukup dekat ke x, maka nilai yang dihasilkan sulit ditebak apakah akan bernilai negatif atau positif karena pada selang yang sangat sempit di sekitar 0 tersebut, fungsi sinus ini berosilasi begitu rapat dan cepat. Perilaku inilah yang akan dimanfaatkan untuk menyeleksi nilai bit yang akan dikeluarkan apakah 1 atau 0 dan disaat yang sama juga menambahkan fitur peningkatan keacakan. Namun, hasil ini hanya dapat diharapkan jika x cukup dekat ke 0. Itulah alasan utama mengapa langkah ke-3 yaitu membagi hasil peta logistik dengan 10 menjadi langkah yang sangat penting.

Karena berbasis *chaos*, nilai inisialisasi x_0 yang dipilih pada langkah 1 akan menghasilkan deretan bit yang jauh berbeda meskipun perbedaan x_0 yang dimasukkan hanya sedikit. Fenomena ini akan dibuktikan pada bagian pengujian. Pada langkah 2, persamaan peta logistik bekerja. Agar tingkat keacakan yang dihasilkan optimal dipilih parameter laju pertumbuhan yang merupakan ujung kanan selang yaitu r = 4. Nilai lelaran berikutnya yaitu x_{r+1} akan sangat sulit diprediksi karena perilaku *chaos* yang dimiliki peta ini. Langkah 3 seperti yang telah dibahas sebelumnya merupakan langkah penting untuk mendekatkan nilai yang dikeluarkan ke nilai 0. Hal ini dilakukan agar nilai acak masuk ke dalam 'area osilasi tinggi' pada fungsi sin(1/x3) yang akan bertugas melakukan pendiskritan. Area osilasi tinggi ini terdapat pada interval yang dekat ke 0. Jika nilai acak yang akan didiskritkan masuk dalam area ini, diharapkan hasil yang didapat tidak hanya menjadi diskrit namun juga dapat mempunyai tingkat keacakan lebih tinggi. Pada langkah 4 pendiskritan dilakukan, hasil keluaran dari fungsi sinus osilasi tinggi di langkah ini benar-benar sulit diprediksi karena osilasi yang sangat rapat akan membuat perubahan hasil yang sangat cepat. Hal ini dimanfaatkan pada langkah 5 untuk membangkitkan bilangan bit 1 atau 0. Jika hasil yang didapat bernilai lebih besar atau sama dengan 0 maka dibangkitkan bit 1, selain itu akan dibangkitkan bit 0. Tentu karena kecepatan osilasi yang dimiliki fungsi sinus di langkah 4, bit yang dihasilkan juga akan sulit diprediksi. Seperti cara iteratif peta *chaos* pada umumnya, langkah pembangkitan terus diulang sampai jumlah iterasi yang diinginkan. Tiap iterasi akan menghasilkan bit 1 atau bit 0. Hasil akhir setelah semua iterasi selesai tentu saja adalah sebuah deretan bit acak yang akan dibuktikan lolos uji keacakan.

Dari segi analisis kompleksitas metode yang diusulkan, metode pendiskritan ini hanya menambahkan tingkat kerumitan dari segi komputasi pada bagian penghitungan $\sin(1/x^3)$. Pada setiap lelarannya harus dilakukan penghitungan fungsi sinus ini. Jika jumlah iterasi tidak banyak, ini mungkin tak terlalu berpengaruh, namun pada

jumlah iterasi yang banyak komputasi sinus akan menjadi usaha komputasi yang berat. Dari segi program, metode pendiskritan ini tidak menambah kerumitan. Ini terlihat kontras pada kode semu fungsi logistik yang sudah ditambahkan dengan pendiskritan sebagai berikut :

```
ArrayInt flogistik(x : real,iterasi : integer)

KAMUS LOKAL
      i : integer
      hasil : integer
      ArrHsl : Array[1..iterasi] of integer
      Y : real

PROGRAM

      for (i = 1 to iterasi)

            x = r * x * (1 - x);

            x= x/10;
            y= sin(1/x^3);

            if (y >= 0) then

                  hasil = 1

            else

                  hasil = 0
            end if

            ArrHsl[i] = hasil
      end for

      return ArrHsl

}
```

Gambar 3. Algoritma pendiskritan (dalam kotak biru) yang ditambahkan pada kode semu peta logistik

Program di atas memperlihatkan implementasi kode semu peta logistik yang sudah dilengkapi dengan fitur pendiskritan. Kotak biru memperlihatkan algoritma komputasi yang ditambahkan untuk melakukan tugas pendiskritan.

Implementasi tersebut membuktikan bahwa tingkat kesulitan pemrograman tidak terlalu meningkat dengan adanya pendiskritan, namun kompleksitas komputasi bertambah dengan adanya penghitungan fungsi sinus di setiap lelaran.

IV. EKSPERIMEN DAN HASIL

A. Pengantar Uji Keacakan Statistik

Uji keacakan statistik digunakan untuk menguji tingkat keacakan dari sebuah deretan bilangan yang dihasilkan pembangkit bilangan acak. Sebuah deretan bilangan acak akan diperiksa dan dihitung properti statistiknya untuk menentukan apakah deretan tersebut lolos uji sebagai bilangan acak atau tidak. Uji keacakan statistik yang akan digunakan untuki pengujian metode pembangkitan bilangan acak di dalam makalah ini adalah *Runs Test*.

Prinsip *Runs Test* adalah memantau kemunculan run yaitu bit-bit berurutan yang bernilai sama untuk menentukan apakah osilasi perubahan bit 1 dan 0 terlalu cepat atau terlalu lambat [4].

Hasil dari *Runs Test* adalah sebuah P-value. Jika nilai p kurang dari 0.01 maka deretan bilangan tidak lulus uji dan tidak dapat dikatakan random. Jika nilai p lebih besar atau samadengan 0.01 maka deretan bilangan lulus uji sebagai bilangan acak [5]. Secara umum semakin besar p-value hasil dari *Runs Test* semakin baik tingkat keacakan dari sebuah deretan bilangan. Parameter ini akan menjadi dasar dalam pengujian hasil bilangan acak dari metode yang diusulkan pada makalah ini dan juga pembandingannya dengan deret bilangan dari pembangkit bilangan acak tanpa pendiskritan.

B. Eksperimen Pembangkit Bilangan Yang Diusulkan

Pada bagian ini akan dilakukan pengujian pembangkitan bilangan acak menggunakan peta logistik yang dilengkapi dengan pendiskritan menggunakan fungsi trigonometri osilasi tinggi. Semua pengujian dilakukan dengan perangkat lunak MATLAB R2008a. *Runs Test* dilakukan dengan bantuan Statistic Toolbox MATLAB. Pada eksperimen pertama, akan dilakukan pengujian pembangkitan dengan dua buah nilai inisialisasi awal yang sedikit berbeda untuk melihat kepekaan dari pembangkit bilangan acak yang diusulkan. Eksperimen kedua adalah pengujian *Runs Test* pada pembangkit bilangan untuk menentukan kelayakan pembangkit ini sebagai pembangkit bilangan acak. Eksperimen ketiga adalah pembandingan hasil *Runs Test* pembangkit bilangan acak *logistic map* tanpa pendiskritan dengan dengan hasil *Runs Test* pembangkit bilangan acak *logistic map* dengan pendiskritan menggunakan fungsi trigonometri osilasi tinggi. Eksperimen ketiga ini dilakukan untuk melihat apakah fungsi sinus berosilasi tinggi pada pendiskritan dapat meningkatkan tingkat keacakan hasil deretan bilangan peta logistik.

Eksperimen 1 : Uji Kepekaan Perubahan Parameter

Pada eksperimen ini, dilakukan pengujian terhadap Pembangkit Bilangan Acak dengan pendiskritan pada dua parameter inisialisasi awal yang memiliki perbedaan sebesar 0.0001. Eksperimen dilakukan untuk melihat kepekaan keacakan terhadap perubahan parameter.

Tabel 1. Pembangkitan bit acak dengan inisialisasi awal x_0 = 0.5634

Inisialisasi awal x_0 = 0.5634, Jumlah iterasi = 56														
1	1	0	1	0	1	1	0	0	1	0	0	0	1	
1	0	0	0	0	1	1	1	1	1	1	0	1	0	
0	1	0	0	1	0	1	0	1	1	0	0	1	1	
0	1	1	0	0	1	0	1	1	0	0	0	0	0	

Tabel 2. Pembangkitan bit acak dengan inisialisasi awal x_0 = 0.5633

Inisialisasi awal x_0 = 0.5633, Jumlah iterasi = 56
1 0 0 0 0 1 0 1 1 1 0 1 1 1 1 0 1 1 1 1 0 1 1 1 0 0 0 1 1 0 1 0 0 0 1 1 0 0 1 0 0 1 1 0 0 1 0 0 1 0 0 0 0 1 0 1

Hasil eksperimen ini menunjukkan bahwa pembangkit bilangan acak *logistic map* dengan pendiskritan menggunakan fungsi trigonometri osilasi tinggi memiliki kepekaan pembangkitan yang sangat tinggi terhadap perubahan nilai inisialisasi awal yang kecil sekalipun. Dua nilai inisialisasi awal yang memiliki perbedaan nilai yang kecil akan menghasilkan dua buah deretan bilangan bit acak yang sangat berbeda. Ini menunjukkan bahwa pendiskritan yang ditawarkan tidak mengubah sifat alami dari pembangkit bilangan acak *logistic map* yang bersifat *chaotic*.

Eksperimen 2 : Uji Keacakan Statistik Runs Test

Pada eksperimen ini, dilakukan uji keacakan statistik *Runs Test* terhadap pembangkit bilangan acak peta logistik dengan pendiskritan fungsi trigonometri osilasi tinggi. Eksperimen *Runs Test* dilakukan untuk menentukan apakah hasil dari pembangkitan bilangan acak dengan pendiskritan yang diusulkan dalam makalah ini layak atau tidak sebagai deretan bilangan random. Jika lulus, yaitu saat nilai p lebih besar dari 0.01, maka deretan bit yang dihasilkan dinyatakan lulus uji keacakan. Untuk memastikan bahwa metode yang diusulkan benar-benar lulus uji, maka dilakukan *Runs Test* terhadap deretan bilangan dari tiga parameter nilai inisialisasi awal yang berbeda.

Tabel 3. Pengujian *Runs Test* pada pembangkitan bilangan acak dengan inisialisasi awal x_0 = 0.6439

Inisialisasi awal x_0 = 0.6439, Jumlah iterasi = 70
1 1 0 0 0 1 1 1 0 1 0 1 1 1 1 1 1 0 0 0 0 1 1 1 1 1 0 0 0 0 1 1 1 1 0 1 1 0 1 0 1 1 0 0 1 0 0 0 0 1 0 1 1 1 1 0 1 0 0 1 0 1 0 0 0 1 1 0 0 1
Hasil *Runs Test* : p-value = 0.5858 > 0.01 Kesimpulan : Lolos uji keacakan

Hasil uji keacakan *Runs Test* menunjukkan bahwa nilai p yang dihasilkan dari deretan bilangan dari pembangkit bilangan peta logistik dengan pendiskritan yang diusulkan lebih besar dari 0.01, sehingga secara statistik metode ini lulus uji keacakan statistik sebagai pembangkit bilangan acak. Untuk mamastikan hasil pengujian, hasil *Runs Test* pembangkit ini akan dilakukan dua kali lagi dengan nilai inisialisasi awal yang berbeda :

Tabel 4. Pengujian *Runs Test* pada pembangkitan bilangan acak dengan inisialisasi awal x_0 = 07793

Inisialisasi awal x_0 = 0.7793 Jumlahiterasi = 70
0 1 0 0 1 1 1 1 1 1 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 1 1 1 1 1 0 1 1 1 1 0 1 0 1 0 0 1 1 1 0 0 1 0 1 0 0 1 0 0 1 1 1 1 1 0 1 1 0 0 1 0 1 1 1 1
Hasil *Runs Test* : p-value = 0.4322 > 0.01 Kesimpulan : Lolos uji keacakan

Tabel 5. Pengujian *Runs Test* pada pembangkitan bilangan acak dengan inisialisasi awal x_0 = 0.8159

Inisialisasi awal x_0 = 0.8159 Jumlah iterasi = 70
0 1 0 0 0 1 1 0 1 1 1 1 0 0 0 0 0 1 1 0 0 0 1 1 0 1 1 0 1 1 0 0 0 1 0 0 1 1 0 1 0 0 0 1 1 1 0 0 1 1 1 1 1 0 1 1 1 1 1 1 1 0 1 1 1 1 0 1 1 0
Hasil *Runs Test* : p-value = 0.3521 > 0.01 Kesimpulan : Lolos uji keacakan

Uji coba dengan semua parameter inisialisasi yang diujicobakan menunjukkan nilai p lebih besar dari 0.01. Hasil ini mengindikasikan bahwa pembangkitan bilangan acak dengan pendiskritan menggunakan fungsi trigonometri osilasi tinggi lulus dari segi keacakan.

Eksperimen 3 : Uji Perbandingan Tingkat Keacakan Pembangkit Bilangan Peta Logistik Dengan dan Tanpa Pendiskritan

Eksperimen ini dilakukan untuk membandingkan tingkat keacakan deretan bilangan yang dihasilkan oleh pembangkit bilangan peta logistik dengan dan tanpa pendiskritan menggunakan fungsi sinus osilasi tinggi yang diusulkan makalah ini. Tujuan yang ingin dicapai adalah untuk melihat apakah metode pendiskritan menggunakan fungsi trigonometri osilasi tinggi dapat meningkatkan keacakan dari pembangkit bilangan peta logistik biasa. Parameter yang dibandingkan pada eksperimen ini adalah nilai p. Jika nilai p semakin besar,

maka tingkat keacakan deretan bilangan semakin baik.

Pembandingan dilakukan pada parameter-parameter inisialisasi awal yang sama dengan parameter pada eksperimen uji keacakan.

Tabel 6. Perbandingan tingkat keacakan pembangkitan bilangan acak peta logistik dengan dan tanpa pendiskritan pada nilai inisialisasi awal $x_0 = 0.6439$

Inisialisasi awal $x_0 = 0.6439$ Jumlah iterasi = 70	
Peta logistik biasa	Hasil *Runs Test* : p-value = 0.0216
Peta logistik dengan pendiskritan menggunakan fungsi trigonometri osilasi tinggi	Hasil *Runs Test* : p-value = 0.5858

Tabel 7. Perbandingan tingkat keacakan pembangkitan bilangan acak peta logistik dengan dan tanpa pendiskritan pada nilai inisialisasi awal $x_0 = 0.7793$

Inisialisasi awal $x_0 = 0.7793$ Jumlah iterasi = 70	
Peta logistik biasa	Hasil *Runs Test* : p-value = 0.1875
Peta logistik dengan pendiskritan menggunakan fungsi trigonometri osilasi tinggi	Hasil *Runs Test* : p-value = 0.4322

Tabel 8. Perbandingan tingkat keacakan pembangkitan bilangan acak peta logistik dengan dan tanpa pendiskritan pada nilai inisialisasi awal $x_0 = 0.8159$

Inisialisasi awal $x_0 = 0.8159$ Jumlah iterasi = 70	
Peta logistik biasa	Hasil *Runs Test* : p-value = 0.2643
Peta logistik dengan pendiskritan menggunakan fungsi trigonometri osilasi tinggi	Hasil *Runs Test* : p-value = 0.3521

Dari hasil pembandingan, didapatkan bahwa nilai p dari bilangan acak yang dihasilkan oleh peta logistik dengan pendiskritan selalu lebih besar dibandingkan dengan nilai p dari bilangan acak peta logistik tanpa pendiskritan. Hasil ini menunjukkan bahwa pendiskritan menggunakan fungsi trigonometri osilasi tinggi berhasil meningkatkan tingkat keacakan dari peta logistik. Metode pendiskritan yang diusulkan terbukti dapat menghasilkan tingkat keacakan bilangan yang lebih baik.

V. ANALISIS DAN PERBANDINGAN

Eksperimen yang dilakukan terhadap pembangkit bilangan acak menunjukkan bahwa metode pendiskritan yang diusulkan dapat menghasilkan deretan bilangan yang lulus uji keacakan. Deretan bilangan yang dihasilkan juga memiliki tingkat keacakan yang lebih baik dibandingkan bilangan dari peta logistik tanpa pendiskritan. Peningkatan keacakan disebabkan oleh penggunaan fungsi trigonometri osilasi tinggi dalam langkah pendiskritan. Meskipun dilakukan penambahan langkah untuk pendiskritan, eksperimen kepekaan menunjukkan bahwa sifat *chaos* tetap dipertahankan dalam metode pembangkitan peta logistik dengan fungsi trigonometri yang diusulkan ini.

Peningkatan manfaat yang didapatkan setelah penambahan fitur pendiskritan menggunakan fungsi trigonometri osilasi dirangkum sebagai berikut :

- ➔ Dapat mengubah bentuk keluaran dari deretan bilangan real menjadi deretan bit sehingga dapat digunakan untuk aplikasi-aplikasi kriptografi.
- ➔ Dapat meningkatkan keacakan deretan bilangan dengan penambahan penghitungan fungsi trigonometri berosilasi tinggi.

Hasil yang didapatkan pada metode pembangkitan fungsi trigonometri muncul dalam bentuk deretan bit 1 atau 0. Format hasil ini cocok digunakan untuk aplikasi-aplikasi kriptografi. Biasanya format hasil seperti ini terdapat pada pembangkit-pembangkit bilangan acak yang dirancang khusus untuk kriptografi dan bersifat *Cryptographically Secure* seperti pembangkit bilangan acak Blum-Blum Shub. Namun dengan metode pendiskritan yang ditawarkan dalam makalah ini, basis teori *chaos* pun dapat digunakan secara maksimal untuk membangkitkan deretan bilangan dalam format bit dengan tingkat keacakan yang tinggi.

Pembangkit bilangan acak dikatakan *Cryptographically Secure* dengan salah satu syarat : harus lulus uji keacakan statistik. Berdasarkan eksperimen yang telah dilakukan terhadap pembangkit bilangan acak peta logistik dengan pendiskritan menggunakan fungsi trigonometri osilasi tinggi, metode yang diusulkan ini juga terbukti menghasilkan deretan bilangan acak yang selalu lulus dalam uji keacakan. Perbedaannya, metode yang diusulkan dalam makalah ini baru terbukti lulus untuk satu jenis pengujian saja yaitu *Runs Test*.

Perbedaan lain antara metode pembangkitan dalam makalah ini dengan metode pembangkitan *Cryptographically Secure* adalah proses komputasi yang dilakukan. Metode pembangkitan peta logistik yang ditawarkan makalah ini bekerja dalam operasi-operasi bilangan real, sedangkan metode pembangkitan *Cryptographically Secure* melakukan operasi-operasi dalam bilangan bulat. Meskipun begitu, keduanya tetap menghasilkan deretan bilangan yang berformat bit.

Berikut adalah perbandingan metode pembangkitan

peta logistik dengan pendiskritan menggunakan fungsi osilasi tinggi dengan pembangkitan bilangan acak *Cryptographically Secure* :

Tabel 9. Perbandingan pembangkit bilangan acak peta logistik pendiskritan dengan pembangkit bilangan acak *Cryptographically Secure* dari berbagai aspek

	Pembangkit Peta Logistik dengan Pendiskritan Menggunakan Fungsi Trigonometri Osilasi Tinggi	**Pembangkit Bilangan Acak *Cryptographically Secure***
Bentuk keluaran	Deretan bit	Deretan bit
Operasi komputasi	Operasi bilangan real	Operasi bilangan bulat
Kompleksitas komputasi	Agak kompleks dengan adanya operasi bilangan real	Sederhana, karena hanya menjalankan operasi-operasi bilangan bulat seperti modulo
Keacakan	Terbukti lulus uji keacakan *Runs Test*	Terbukti lulus semua uji keacakan statistik

Secara komputasi pembangkit bilangan acak lebih sederhana untuk diselesaikan. Meskipun demikian pembangkit peta logistik dengan pendiskritan yang diusulkan dalam makalah ini merupakan pendekatan yang baru untuk menghasilkan deretan bit dengan keacakan tinggi dari kelas pembangkit bilangan berbasis *chaos*.

VI. KESIMPULAN

Pada makalah ini telah diusulkan sebuah metode pendiskritan hasil dari pembangkit bilangan acak peta logistik dengan menggunakan penghitungan fungsi trigonometri osilasi tinggi. Fungsi trigonometri yang dipilih adalah yang dipilih adalah $y=\sin(1/x^3)$ yang memiliki frekuensi sangat tinggi pada parameter x yang dekat dengan 0. Hasil bilangan acak peta logistik yang masih berbentuk real diseleksi dengan menjadikannya parameter untuk penghitungan fungsi $y=\sin(1/x^3)$. Sebelumnya interval hasil peta logistik dipersempit terlebih dahulu menjadi [0,0.1] agar sedekat mungkin dengan nilai 0 sehingga hasil yang didapatkan masuk pada daerah osilasi tinggi dari fungsi sinus. Penyeleksian dilakukan dengan memanfaatkan tanda positif dan negatif dari hasil komputasi fungsi trigonometri yang sulit ditebak. Metode ini berhasil membangkitkan bilangan acak dalam bentuk bit sehingga dapat berguna dalam aplikasi kriptografi. Dari hasil eksperimen, deretan bit yang dihasilkan lulus dalam uji keacakan statistik *Runs Test*. Deretan bit yang didapatkan juga memiliki keacakan yang lebih baik dibandingkan pembangkit peta logistik tanpa pendiskritan. Peningkatan kegunaan tersebut dapat dicapai tanpa menghilangkan karakteristik *chaos* yang dimiliki peta logistik.

REFERENCES

[1] Munir, Rinaldi, *Kriptografi*. Penerbit Informatika, 2007.

[2] Rao, Suresh, SC Phatak "*Logistic map* : A Possible Random Generator", Physical Review, Vol. 5, No. 4, submitted for publication

[3] http://www.ptc.com/appserver/wcms/standards/textoimgothumb.jsp?&im_dbkey=62636&icg_dbkey=888
Tanggal Akses : 26 April 2011 22.40 WIB

[4] http://csrc.nist.gov/groups/ST/toolkit/rng/documents/SP800-22b.pdf
Tanggal Akses : 24 April 2011 10.39 WIB

[5] http://www.fi.muni.cz/~xkrhovj/lectures/2005_PA168_Statistical_Testing_slides.pdf
Tanggal Akses : 26 April 2011 22.23 WIB

[6] http://www.egwald.ca/nonlineardynamics/logisticsmap*chaos*.php
Tanggal Akses : 7 Mei 2011 13.45 WIB

[7] http://www.math.washington.edu/~conroy/general/sin1overx/
Tanggal Akses : 7 Mei 2011 13.53 WIB

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, 9 Mei 2011

ttd

Achmad Dimas Noorcahyo
13508076