

Pembangkit Bilangan Acak dengan Menggunakan *De Jong Attractor* dan Perbandingan dengan Fungsi $\sin(1/x)$

Septu Jamasoka (13509080)
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganessa 10 Bandung 40132, Indonesia
13509080@std.stei.itb.ac.id

Abstract—Bilangan acak sangat penting dalam kriptografi karena sangat mendukung keberhasilan dari sebuah metode kriptografi. Bilangan acak cukup banyak digunakan dalam kriptografi karena memungkinkan diperolehnya hasil *ciphertext* yang acak karena adanya bilangan acak tersebut yang tidak bisa ditebak sembarang dan memerlukan *seed* tertentu sehingga hasilnya bisa sesuai. Oleh karena itu, diperlukan sebuah pembangkit bilangan acak yang cukup bagus sehingga tidak mudah ditebak oleh orang luar terkait pola perhitungan bilangan acak sehingga pesan yang dienkripsi tidak dapat dibongkar dengan mudah oleh orang lain. Salah satu metode sehingga nilai acak sulit ditebak adalah dengan menggunakan *De Jong Attractor*. *Attractor* adalah alat dalam matematika yang digunakan untuk menggambarkan fenomena alam. *De Jong Attractor* merupakan suatu fungsi yang dapat menghasilkan bilangan acak yang bergantung pada empat buah nilai masukan pengguna yang akan menghasilkan dua buah nilai. Fungsi lainnya yaitu $\sin(1/x)$ merupakan fungsi sederhana yang juga menghasilkan nilai acak terutama jika nilai masukan pengguna cukup kecil sekali. Kedua metode ini akan dibandingkan dari segi hasil pembangkitan nilai acak sebanyak n kali secara berulang kali untuk melihat perbandingan bilangan-bilangan acak yang dihasilkan terhadap setiap perulangan untuk kedua metode tersebut.

Index Terms—bilangan acak, pembangkit bilangan acak, *De Jong Attractor*, fungsi $\sin(1/x)$.

I. PENDAHULUAN

Kriptografi sangat berguna sekali untuk menjamin keamanan dalam berkomunikasi antar dua entitas (contohnya manusia, proses, dan lain-lain) atau lebih yang memungkinkan penyembunyian pesan sehingga tidak memungkinkan pesan untuk dibaca oleh orang-orang yang tidak berhak untuk membaca pesan tersebut. Penyembunyian pesan ini tentunya membutuhkan teknik enkripsi yang bagus sehingga menyulitkan orang dalam memecahkan pesan yang disembunyikan. Oleh karena itu, dibutuhkanlah sebuah pembangkit bilangan acak yang bagus sehingga enkripsi pesan juga akan membuat orang semakin sulit memecahkannya.

Akan tetapi, saat ini sudah banyak sekali jenis-jenis serangan yang ditemukan untuk memecahkan enkripsi-enkripsi yang ada dan ditambah dengan kemampuan komputasi yang meningkat dari tahun ke tahun. Hal ini tentunya memaksa banyak pihak yang membutuhkan

kanal yang aman untuk berkomunikasi semakin memikirkan cara yang terbaik untuk menghasilkan enkripsi yang aman. Pembangkit bilangan acak yang ada saat ini terutama yang digunakan sebagai bagian dari enkripsi pesan banyak masih berupa pembangkit bilangan acak semu. Pembangkit bilangan acak semu menghasilkan bilangan-bilangan yang memungkinkan timbul perulangan setelah beberapa kali terjadi perulangan dihasilkan. Hal ini tentunya cukup berbahaya karena dengan melihat pola yang ada, orang-orang dapat memecahkan rumus yang digunakan untuk membuat bilangan acak tersebut.

Oleh karena itu, pembangkit bilangan acak murni sangat diperlukan sehingga pengguna merasa aman. Hal ini karena pembangkit bilangan acak murni tidak memungkinkan ditemukannya pengulangan setelah beberapa kali perulangan dilakukan. Akan tetapi, untuk membuat pembangkit bilangan acak murni, dibutuhkan fungsi *chaos* sebagai rumus dasar dalam pembangkitan bilangan tersebut. Fungsi *chaos* menyebabkan hasil perhitungan berikutnya tidak bisa ditebak dan tidak mungkin adanya pola ataupun perulangan.

De Jong Attractor dan fungsi $\sin(1/x)$ merupakan contoh fungsi *chaos* yang digunakan pada makalah ini untuk dilakukan pengujian dengan *De Jong Attractor* merupakan fungsi *chaos* dua dimensi sehingga diperlukan metode khusus sehingga bekerja dengan baik.

II. DASAR TEORI

A. Pembangkit Bilangan Acak

Bilangan acak adalah sebuah bilangan yang dibangkitkan dari sebuah proses, dengan hasil yang tidak dapat diprediksi, dan tidak memungkinkan juga untuk direproduksi. Alat yang digunakan untuk membuat bilangan acak disebut sebagai pembangkit bilangan acak^[10].

Pembangkit bilangan acak atau Random Number Generator (RNG) merupakan sebuah sistem dengan kemampuan untuk menghasilkan serangkaian bilangan acak berdasarkan fungsi tertentu^[9]. Kebanyakan komputer yang ada sekarang ini hanya bisa menghasilkan bilangan yang bisa ditebak dan terkadang terlihat fenomena periode terjadi sehingga biasa disebut sebagai pembangkit

bilangan acak semu. Pembangkit bilangan acak yang dikatakan baik adalah jika memenuhi syarat berikut^[1].

1. Rangkaian bilangan acak harus memenuhi contoh distribusi yang homogen.
2. Periode dari rangkaian bilangan acak harus cukup panjang sehingga tidak bisa ditebak.
3. Harus efisien dalam pembangkitan bilangan acak tersebut.

Dengan syarat-syarat tersebut, terlihat bahwa pembangkit bilangan acak sangat ditentukan dari segi rangkaian yang dihasilkan serta waktu yang diperlukan.

Pembangkit bilangan acak dapat terdiri dari dua jenis: perangkat lunak dan perangkat keras. Pembangkit bilangan acak yang berupa perangkat lunak biasanya hanya berupa pembangkit bilangan acak semu karena algoritma matematika yang diterapkan pada komputasi memungkinkan terdapatnya periodisasi pada hasil yang ditimbulkan. Pembangkit bilangan acak yang berasal dari perangkat keras memungkinkan pembangkitan bilangan acak yang murni. Akan tetapi, pada perangkat keras memungkinkan timbulnya bias akan ketidakmampuan untuk menghasilkan proses secara fisik yang merata^[2].

B. Chaos Function

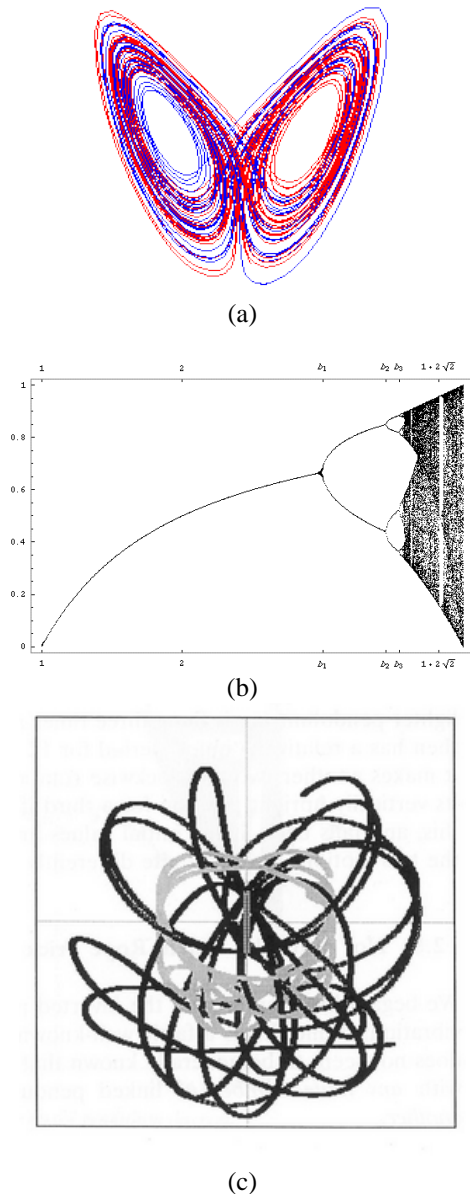
Chaos adalah suatu kondisi dari sistem yang tidak memiliki keteraturan dalam sistem atau dengan kata lain tidak dapat ditemukan suatu pola yang dapat ditentukan dengan pasti^[1]. Teori *chaos* adalah studi dalam bidang fisika, matematika, dan lain-lain yang mempelajari bagaimana sistem dinamis sangat dipengaruhi oleh kondisi awal, dengan kondisi awal yang berbeda sedikit dapat menghasilkan rangkaian bilangan yang sangat beragam antara satu dengan yang lainnya. Sistem yang terbentuk dalam teori *chaos* ini disebut sebagai sistem *chaos*. Contoh-contoh dari sistem *chaos* banyak berupa sistem pada dunia nyata seperti ramalan cuaca, dan lain sebagainya^[3].

Fungsi *chaos* adalah fungsi yang membangkitkan kondisi yang tidak beraturan, yaitu dengan nilai yang awal yang berbeda sedikit, hasil yang diperoleh akan berbeda cukup besar. Contoh fungsi *chaos* yang sudah ditemukan antara lain berupa fungsi logistik (*logistic mapping*), *Lorenz Attractor* atau *strange attractor* yang ditemukan oleh Lorenz dan biasa disebut sebagai *butterfly effect* dan merupakan fungsi *chaos* yang ditemukan pertama kali, *double pendulum*, dan masih banyak lainnya^[7]. Adapun pemetaan atas beberapa hasil fungsi chaos dapat dilihat pada gambar 1.

C. De Jong Attractor

Attractor adalah entitas-entitas matematika (alat, mesin, dan lain-lain) yang digunakan untuk menangani fenomena alami seperti iklim, kimia, dan lain-lain^[6]. Dalam pengertian ini dijelaskan bahwa *attractor* merupakan pencitraan atau penggambaran dari kondisi alami yang cenderung bersifat *chaos*. *De Jong Attractor* merupakan salah satu contoh dari *attractor* berdimensi dua. *De Jong Attractor* ditemukan oleh Peter De Jong

yang menggunakan fungsi trigonometri sebagai dasar dari fungsi *chaos* yang diciptakan. Dalam hal ini, fungsi trigonometri merupakan fungsi yang memiliki range nilai dari -1 hingga +1. Akan tetapi, nilai yang diperoleh cenderung sangat beragam dan bergantung pada nilai awal masukan hingga selisih sedikit (sebagai contoh 0,0001 dengan 0,00001), hasil yang ditimbulkan bisa selisih cukup besar.



Gambar 1. (a) *Lorenz Attractor*, (b) *Logistik Mapping*, (c) *double pendulum*^[7]

Rumusan dari *De Jong Attractor* adalah sebagai berikut.

$$x_{n+1} = \sin(a \times y_n) - \cos(b \times x_n) \dots (2.1)$$

$$y_{n+1} = \sin(c \times x_n) - \cos(d \times y_n) \dots (2.2)$$

Dengan x_{n+1} dan y_{n+1} merupakan nilai x dan y yang baru, a , b , c , dan d merupakan suatu nilai acak yang ditentukan, dan x_n dan y_n merupakan nilai x dan y sebelumnya. Gambar 2 akan menunjukkan beberapa hasil dari *De Jong Attractor* dengan nilai a , b , c , dan d yang dispesifikasikan.

III. IMPLEMENTASI

A. Lingkungan Implementasi

Implementasi yang dilakukan adalah percobaan pembuatan pembangkit bilangan acak dengan menggunakan *De Jong Attractor* dan fungsi $\sin(1/x)$ akan diimplementasikan dengan menggunakan bahasa Java dengan menggunakan *Integrated Development Environment (IDE)* berupa NetBeans 7.1.1 pada sistem operasi Windows © 7. Dalam pembuatannya, program menggunakan *interface* berupa *command*, tanpa menggunakan *graphical user interface* karena tidak diperlukan. Selain itu, masukan *seed* awal ditentukan langsung oleh pengguna dengan memasukkan nilai pada *command prompt*.

Adapun *seed-seed awal* yang dijadikan *seed* percobaan dengan nilai yang berbeda-beda dapat dilihat pada Tabel 1. Dalam percobaan ini, nilai perulangan juga diatur cukup besar yaitu dengan nilai $n = 1000$ agar terlihat apakah terdapat periodisasi yang terjadi dan untuk melihat juga perbedaan waktu untuk memroses antara menggunakan *De Jong Attractor* dengan fungsi $\sin(1/x)$.

Tabel 1. *Seed Awal Percobaan*

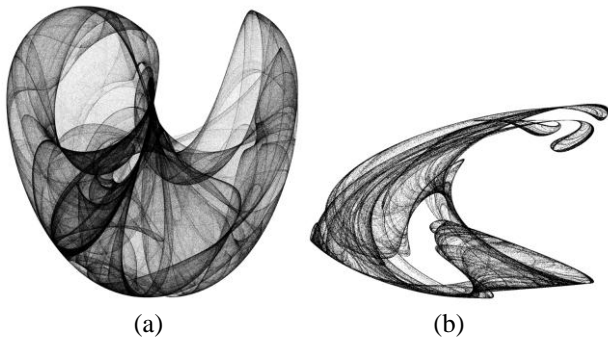
No Percobaan	Seed Awal
1	5435
2	4537734
3	8475892443
4	833245835934
5	734685129382434

B. Implementasi De Jong Attractor

Seperti yang sudah dijelaskan pada II.C., *De Jong Attractor* merupakan salah satu fungsi *chaos* berdimensi dua. Oleh karena itu, implementasi yang diterapkan untuk percobaan adalah dengan memasang nilai x dan y sebagai pasangan hasil bilangan acak dan hasil itu disimpan sementara sebagai masukan untuk perhitungan nilai berikutnya.

Selain itu, pada *De Jong Attractor* dibutuhkan juga nilai a , b , c , dan d . Nilai-nilai tersebut diperoleh dari *seed* awal untuk percobaan yang dibagi menjadi empat bagian sehingga diperoleh nilai a , b , c , dan d . Untuk nilai awal x dan y juga di ambil dari *seed* yang berukuran 8 byte sehingga 4 byte awal untuk x dan 4 byte sisanya untuk y .

Setelah diperoleh nilai a , b , c , dan d serta nilai x dan y , akan dihitung nilai x dan y baru berdasarkan rumus (2.1) dan (2.2) pada bagian II.C. Akan tetapi, hasil yang diperoleh akan berupa bilangan antara -1 hingga 1 yang merupakan *floating point*. Oleh karena itu, karena *floating point* merupakan 8 byte dalam implementasi komputer, 8 byte tersebut diubah menjadi bertipe *long 8 byte* yang di-*reverse* secara byte sehingga dapat dibaca dalam bentuk *integer* (bukan *floating point*). Penyimpanan semua data dalam bentuk *array of bytes* yang memudahkan dalam transformasi untuk perhitungan selanjutnya. Adapun diagram alir secara umum dari implementasi *De Jong Attractor* untuk pembangkit bilangan acak dapat dilihat



Gambar 2. *De Jong Attractor* dengan (a) $a = 1.4, b = -2.3, c = 2.4, d = -2.1$, (b) $a = 2.01, b = -2.53, c = 1.61, d = -0.33$ ^[5]

D. Fungsi $\sin(1/x)$

Fungsi $\sin(1/x)$ juga merupakan salah satu contoh fungsi *chaos*, tetapi hanya untuk x yang kecil saja. Ketika x bernilai semakin besar, maka nilai $\sin(1/x)$ akan cenderung menjadi stabil mendekati nol (0). Oleh karena itu, fungsi $\sin(1/x)$ akan memiliki situasi *chaos* ketika nilai x cukup kecil sekali hal ini karena ketika x semakin mendekati nol (0), nilai $\sin(1/x)$ akan melompat dengan cepat dari +1 ke -1 dan sebaliknya hanya dengan perbedaan yang sedikit. Sebagai contoh misalnya $f = \sin(x)$, maka^[4]

$$f(1 / (10001.5 \pi)) = -1$$

$$f(1 / (10002.5 \pi)) = 1$$

$$f(1 / (10003.5 \pi)) = -1$$

$$f(1 / (10004.5 \pi)) = 1$$

...

$$f(1 / (100000000001.5 \pi)) = -1$$

$$f(1 / (100000000002.5 \pi)) = 1$$

$$f(1 / (100000000003.5 \pi)) = -1$$

$$f(1 / (100000000004.5 \pi)) = 1$$

...

$$f(1 / (1000000000000000000001.5 \pi)) = -1$$

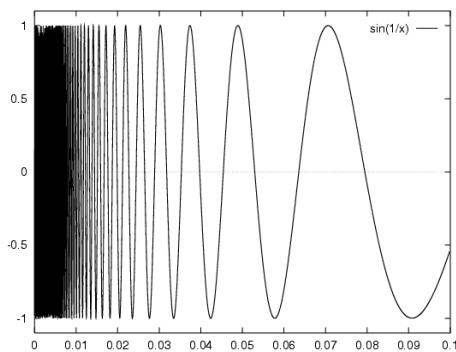
$$f(1 / (1000000000000000000002.5 \pi)) = 1$$

$$f(1 / (1000000000000000000003.5 \pi)) = -1$$

$$f(1 / (1000000000000000000004.5 \pi)) = 1$$

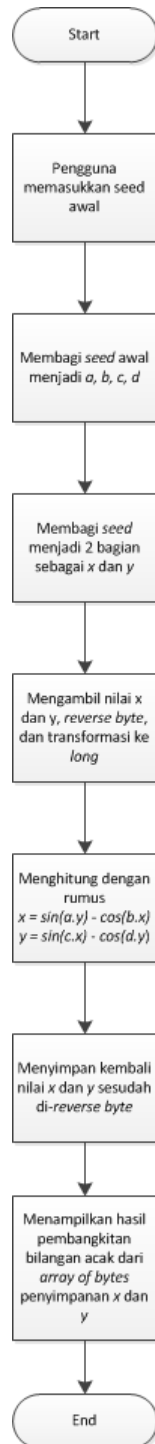
...

dapat terlihat bahwa dengan perbedaan yang sedikit saja, hasilnya akan berubah cukup drastis. *Plot* dari fungsi $\sin(1/x)$ dapat dilihat pada gambar 3.



Gambar 3. Tampilan *Plotting Fungsi* $\sin(1/x)$ ^[4]

pada Gambar 4.



Gambar 4. Diagram Alir Implementasi De Jong Attractor

Hasil uji coba yang dilakukan terhadap implementasi pembangkit bilangan acak dengan De Jong Attractor dapat dilihat pada Tabel 2 berikut.

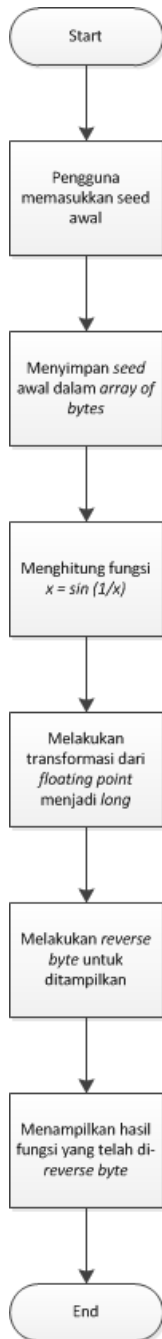
Tabel 2. Hasil Implementasi Pembangkit Bilangan Acak De Jong Attractor dengan Seed Percobaan

No	15 Bilangan Pertama	Waktu (ms)
1	1136908982773270854233791 1136905741465121755425599	202.61835

	1136907805918317029144127 1136898557003187537895743 1136906750266805806163647 1136904783809228900854207 1136902620743349426508863 1136896265708495794986943 1136893947115266554128959 1136904030373608258531135 1136908850466586496788415 1136907668983411412364991 1136907269913977237715775 1136902546663715930498495 1136891293052306497905855	
2	1136900854857140044687679 1136902319426787194499007 1136901382287826646255679 1136893052622409323239999 1136898520594225502017471 1136904466310140773783487 1136892178605376662389055 1136900704372370090882495 1136901441371487101711807 1136891348517453608223295 1136899514166557803340863 1136898884472416317073855 1136896164486572670448063 1136894181866851187619263 1136899215623756190117823	202.05859
3	94455630689180412813072503012537521599 -54849612986756962385305763593421133889 156863336911742269242092364375882855231 72434581062204111011807245948244262079 99376095764953936497933072300946815423 77188960516429881163508312964190815551 -73279095923971800431084384315565868737 96569547874767284846367516829276485823 130825987682348526120176454497819098047 -74981516186059062494864731872779639745 -80055777913724989416887538004472445889 123321399780316393305251787612759126079 -122832724224026279333576066847808954177 107519934863456891136363681988937246015 85735881877663761238600685613694441407	255.45619
4	89405791364040095526088904030737388607 143810369946205666485766987294918244927 -79249465377598856371642731008525206081 65315084029823568334564892860825269311 15411918439686175408946646558388708415 -139857762130006138563435574166735096129 -94313607691439795422924291112478050625 120202052169240989825335847131107929151 -58953363299163323731400550967392281537 22466040875294978556860708169308100287 55084470538067480218183533536544013375 -164698521765151365575462595319140321857 139725626706041250058962065068903101119 -79905553565604004352163545768575828033 133107316933617692159275304443719317823	284.63425
5	-163147389382546554664555897796274751809 101989619586850007325823073108673741631 22174629898780406112075112045716827071 7964626874276029401860928150640196927 101757131544774902294612092515267967807 -62929155487327211884496324357121510465 75794850854514830010988393378670247359 49009601452396258125552445082453140927 34396243106678076714978244180342463423 11583397867312463571007174067949197631 -52552200976908633150788339351171439041 9165871871830077311679202222254584383 117586568263046042784381106748862016447 -32902658942638472922179320634072567105 104675486812134542861334681719076808895	335.39230

C. Implementasi Fungsi Sin(1/x)

Untuk implementasi pembangkit bilangan acak dengan fungsi $\sin(1/x)$ tidak sesulit seperti implementasi *De Jong Attractor*. Nilai x merupakan nilai *seed* yang dimasukkan oleh pengguna. Dalam hal ini, nilai x akan selalu berubah seiring pemanggilan fungsi pembangkit. Nilai x akan disimpan dalam bentuk *array of bytes*. Ketika menampilkan hasilnya, hasil dari fungsi tersebut yang merupakan *floating point* akan diubah ke dalam bentuk *long*, yang kemudian akan di-*reverse byte*. Implementasi dalam diagram alir dapat dilihat pada Gambar 5.



Gambar 5. Diagram Alir Implementasi fungsi Sin(1/x)

Hasil implementasi pembangkit bilangan acak dengan fungsi $\sin(1/x)$ dapat dilihat pada Tabel 3.

Tabel 3. Hasil Implementasi Pembangkit Bilangan Acak Fungsi $\sin(1/x)$ dengan *Seed Percobaan*

No	15 Percobaan Pertama	Waktu (ms)
1	5921551524015450175 -6264408680080668612 4905982222954006460 4257821058365918780 -7259491959732694724 3475693574401819836 3037934795141813564 -3675604446671726532 -8121020107929217860 6145390729237299900 -504371347506132932 -7124371184693984580 1743563980262474940 -8598273816849275588 -212066184447065668	148.760856
2	4817610175358995774 8058170348192992828 -3920846566410157508 -1757668925684772164 -5234618413745363780 -5998720176828117828 -5996072322185230148 904259204491774140 6716357966530229308 5172658860593775932 -805766450994803652 9025611357405591228 1526274321844600892 6528477537321625660 -542144587274516932	170.405362
3	6126557366867976253 8403193285532059708 733178979813359932 -1151009157765646020 6153243277266399420 6840663869174253372 -7577484374957750980 3618836629314667452 1511690030314361916 1421488351380711484 6817982123596851516 8189801978589939004 -5821649662632525252 -7630108420165006916 -3371105681734724 -1162049559872883268	173.350671
4	4441857745428968765 350300941164744764 -8859405979325543876 -7135724841245081412 -3833992358791347012 4499508318133162940 -8018462155938066372 2140678025332064956 2314666980990001468 1213466377703137084 -1164082225085993412 6261198000323899324 6857114732681299772 6490729663995118908 -8791124554887199172	178.222132
5	-1500053263724718020 -7809368355477378884 -8223969329408572740 -1943363421316513348 6073320980455563964 -3405429182702024644 5523904846398887356	180.904713

2493516304640051772	
-7307324731263869636	
278951825195009212	
-3899723715532271556	
7056928151773582012	
508494352782787644	
2605713719449305660	
-3405811392814441412	

IV. ANALISIS

A. Analisis De Jong Attractor

Dari hasil percobaan terhadap pembangkit bilangan acak dengan *De Jong Attractor* yang dibuat pada Tabel 2, diperoleh data bahwa dengan *seed* yang kecil, hasil yang diperoleh bisa ditebak bagian awalnya karena pembagian berdasarkan 4 *byte* yang ditentukan dalam implementasi menyebabkan nilai x selalu tidak berubah dan hanya nilai y yang berubah. Hal ini tentunya mengakibatkan mudah bagi orang luar untuk menebak digit-digit awal dari bilangan acak yang mungkin ditampilkan.

Untuk rangkaian bilangan acak yang dihasilkan dari *seed* yang cukup panjang, hasil yang diperoleh cukup acak dan tidak dapat dilihat polanya terutama dari segi negatif atau positif bilangan berikutnya. Selain itu, panjang bilangan yang cukup panjang menyebabkan orang yang berniat menebak bilangan selanjutnya akan mengalami kesulitan jika tidak menggunakan alat bantu.

Dari percobaan dengan seribu kali perulangan, tidak ditemukan sama sekali perulangan terjadi. Dari hal ini, dapat diketahui bahwa penggunaan *De Jong Attractor* untuk membangkitkan bilangan acak cukup bagus karena tidak mengalami perulangan bahkan hingga seribu bilangan berikutnya.

Dari segi waktu yang diperlukan (berdasarkan Tabel 2), waktu bertambah seiring dengan bertambah panjangnya *seed*. Selain itu, waktu rata-rata yang diperlukan untuk menghasilkan seribu bilangan pertama adalah 256.03193 *ms* atau untuk menghasilkan satu bilangan diperlukan 0.256 *ms*.

B. Analisis Fungsi Sin(1/x)

Berdasarkan Tabel 3 yang merupakan hasil dari percobaan hasil implementasi pembangkit bilangan acak dengan fungsi $\sin(1/x)$, diperoleh bahwa rangkaian bilangan acak yang diperoleh cukup acak baik untuk *seed* dengan digit sedikit hingga *seed* dengan digit yang cukup banyak. Selain itu, pola bilangan acak yang dihasilkan terutama untuk positif dan negatif cukup sulit ditebak karena cukup acak hasil yang ditimbulkan.

Dari percobaan dengan seribu kali perulangan, pembangkit bilangan acak ini juga tidak memperlihatkan kondisi timbulnya perulangan. Hal ini menandakan bahwa pembangkit bilangan acak dengan menggunakan fungsi $\sin(1/x)$ cukup menunjukkan hasil yang memuaskan karena hingga perulangan keseribu masih tidak ditemukan adanya perulangan yang terjadi.

Waktu yang diperlukan juga cukup singkat, yaitu rata-rata waktu yang diperlukan adalah 170.3287 *ms* dengan

untuk menghasilkan sebuah bilangan hanya diperlukan waktu 0.170 *ms*.

C. Perbandingan Implementasi dengan De Jong Attractor dan Fungsi Sin(1/x)

Berdasarkan dua buah percobaan yang sudah dilakukan yaitu implementasi pembangkit bilangan acak dengan *De Jong Attractor* dan dengan fungsi $\sin(1/x)$, terdapat perbedaan-perbedaan yang terlihat dari hasil yang diperoleh. Perbedaan tersebut antara lain berupa fungsi yang digunakan, panjang digit dari bilangan yang dihasilkan, waktu rata-rata yang diperlukan untuk pembangkitan satu buah bilangan, dan jenis *seed* sehingga diperoleh hasil yang cukup baik dapat dilihat pada Tabel 4 berikut.

Tabel 4. Perbandingan antara Implementasi Pembangkit Bilangan Acak dengan De Jong Attractor dan Fungsi Sin(1/x)

Perbedaan	<i>De Jong Attractor</i>	Fungsi $\sin(1/x)$
Fungsi	$x_{n+1} = \sin(a \times y_n) - \cos(b \times x_n)$ $y_{n+1} = \sin(c \times x_n) - \cos(d \times y_n)$	$x_{n+1} = \sin\left(\frac{1}{x_n}\right)$
Panjang digit	20 - 40	20
Waktu per satu bilangan (ms)	0.256	0.170
Panjang seed	Panjang	Pendek

Dari Tabel 4 diatas, *De Jong Attractor* membutuhkan dua buah fungsi sedangkan fungsi $\sin(1/x)$ hanya membutuhkan satu buah fungsi sehingga *De Jong Attractor* akan lebih sulit ditebak dibandingkan dengan fungsi $\sin(1/x)$, tetapi membutuhkan waktu perhitungan lebih. Pembangkit bilangan acak dengan menggunakan *De Jong Attractor* menghasilkan panjang digit bilangan yang lebih daripada fungsi $\sin(1/x)$. Dengan semakin panjangnya bilangan yang dihasilkan tentunya hal ini akan membuat semakin sulit bagi orang lain untuk menebak bilangan berikutnya juga bilangan tersebut semakin panjang. Dari segi waktu, implementasi *De Jong Attractor* membutuhkan waktu lebih lama dibandingkan dengan fungsi $\sin(1/x)$. Hal ini terjadi karena dalam implementasi, *De Jong Attractor* membutuhkan waktu terlebih dahulu untuk memisahkan x dan y , kemudian menghitung dengan menggunakan dua buah rumus. Hal ini menyebabkan *De Jong Attractor* lebih lama dibandingkan dengan fungsi $\sin(1/x)$ yang hanya membutuhkan perhitungan satu rumus untuk menghasilkan bilangan acak. Dari segi panjang kunci yang dibutuhkan, *De Jong Attractor* membutuhkan *seed* dengan panjang digit yang cukup panjang sedangkan untuk fungsi $\sin(1/x)$ justru dibutuhkan *seed* yang pendek karena semakin pendek, hasil fungsi $\sin(1/x)$ akan semakin *chaos*, sedangkan jika semakin panjang *seed*, maka hasil dari fungsi $\sin(1/x)$ akan cenderung mendekati nilai nol.

V. KESIMPULAN

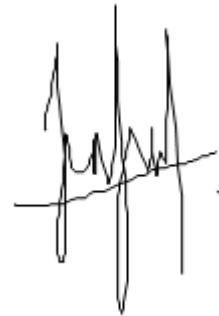
Dari hasil implementasi pembangkit bilangan acak yang dilakukan, *De Jong Attractor* dan fungsi $\sin(1/x)$ dapat digunakan sebagai dasar algoritma untuk membangkitkan bilangan acak karena keduanya merupakan fungsi *chaos* dan cukup baik karena tidak terdapat periodisasi hingga percobaan dengan perulangan dilakukan hingga seribu kali. Diantara kedua metode tersebut baik yang menggunakan *De Jong Attractor* ataupun dengan menggunakan fungsi $\sin(1/x)$, masing-masing memiliki kekurangan dan kelebihan masing-masing. *De Jong Attractor* menghasilkan panjang bilangan acak yang lebih panjang yang tentunya akan menghasilkan keamanan lebih baik dibandingkan dengan menggunakan fungsi $\sin(1/x)$. Akan tetapi, fungsi $\sin(1/x)$ hanya membutuhkan *seed* yang pendek untuk menghasilkan hasil yang maksimal dalam membangkitkan nilai acak serta waktu yang lebih cepat, tetapi fungsi untuk pembangkitannya mudah ditebak dibandingkan dengan *De Jong Attractor*.

Apapun metode yang dipilih baik *De Jong Attractor* maupun fungsi $\sin(1/x)$ tentunya merupakan pilihan dari masing-masing pengembang sistem keamanan dan disesuaikan dengan tingkat keamanan yang diharapkan.

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, 12 Mei 2012



Septu Jamasoka
13509080

DAFTAR REFERENSI

- [1] Guan Xiaohui; Qian Yaguan; , "The Design of Combined Random Number Generator," *Multimedia Information Networking and Security (MINES), 2010 International Conference on Multimedia Information Networking and Security*, pp.640-643, 4-6 Nov. 2010. (<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5671018&isnumber=5670801>)
- [2] Bastos-Filho, C.J.A.; Andrade, J.D.; Pita, M.R.S.; Ramos, A.D.; , "Impact of the quality of random numbers generators on the performance of particle swarm optimization," *Systems, Man and Cybernetics, 2009. SMC 2009. IEEE International Conference on Systems, Man, and Cybernetics*, pp.4988-4993, 11-14 Oct. 2009. (<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5346366&isnumber=5345886>)
- [3] <http://blog.nihilogic.dk/2009/10/strange-attractors-beautiful-chaos-and.html>, diakses pada tanggal 12 Mei 2012, pukul 18.52 WIB.
- [4] <http://computing.dcu.ie/~humphrys/Notes/Neural/chaos.html>, diakses pada tanggal 12 Mei 2012, pukul 18.04 WIB.
- [5] <http://local.wasp.uwa.edu.au/~pbourke/fractals/peterdejong/>, diakses pada tanggal 12 Mei 2012, pukul 19.00 WIB.
- [6] <http://www.bentamari.com/attractors.html>, diakses pada tanggal 12 Mei 2012, pukul 18.50 WIB.
- [7] <http://www.clausewitz.com/Complex/ChaosDemos.htm>, diakses pada tanggal 12 Mei 2012, pukul 18.01 WIB.
- [8] <http://www.complexification.net/gallery/machines/peterdejong/>, diakses pada tanggal 12 Mei 2012, pukul 19.01 WIB.
- [9] <http://www.randomnumbers.info/content/Generating.htm>, diakses pada tanggal 12 Mei 2012, pukul 17.32 WIB.
- [10] <http://www.randomnumbers.info/content/Random.htm>, diakses pada tanggal 12 Mei 2012, pukul 17.32 WIB.