ISSN 1693 - 227

# FAHMA

## JURNAL TEKNOLOGI INFORMASI DAN ILMU KOMPUTER

**Volume 13, Nomor 1** **Januari 2015**



Diterbitkan oleh :
Lembaga Penelitian dan Pengabdian Masyarakat
STMIK EL RAHMA YOGYAKARTA

Jurnal FAHMA Volume 13 Nomor 1 Januari 2015

Jurnal FAHMA merupakan jurnal di bidang teknologi informasi dan ilmu komputer beserta rumpun keilmuannya. Diterbitkan oleh LP2M STMIK EL RAHMA dengan frekuensi terbit setahun tiga kali pada bulan Januari, Mei dan September.



**Alamat redaksi:** Unit LP2M (Lembaga Penelitian dan Pengabdian Masyarakat)
STMIK EL RAHMA Jl. Sisingamangaraja No. 76 Yogyakarta
**e-mail:** lp2m@stmikelrahma.ac.id **Telepon/fax:** 0274 – 377982

# DAFTAR ISI

# RANDOM NUMBER GENERATOR DENGAN METODE LINEAR CONGRUENT

**Pulut Suryati [1], FX. Henry Nugroho [2]**

[1,3]Program Studi Sistem Informasi, STMIK AKAKOM Yogyakarta
e-mail: lut_surya@akakom.ac.id1, 2henry@akakom.ac.id

*Abstract*

*Random number or a random number is a number generated from a process, the output is unpredictable and can not be generated sequentially numbers the same. Randon numbers are useful for many purposes, such as generating key data encryption, simulation and modeling of complex phenomena and to select a random sample from a larger data sets. The problem that arises is how not easy to be able to produce numbers that truly random. The process that generates a random number called the random number generator. In this research method used in this paper is to use a linear congruential method using the chi-square test . The aim of this study was to determine how the randomness of the numbers generated from inear method Congruent with the chi-square test. Random number generation algorithm with linear congruential generator method showed that the resulting number is random, but based on a use of the chi-square test of linear congruential generator methods still are looping validation it is shown that the random number is false. Determination of constants LCM (a, c and m) largely determines whether or not the random numbers obtained.*

*Keywords—3-5random number, LCM, chi-square*

## PENDAHULUAN

Saat ini bilangan acak banyak gunakan untuk berbagai keperluan dan aplikasi, seperti menghasilkan kunci enkripsi data, simulasi dan pemodelan fenomena kompleks dan untuk memilih sampel acak dari data set yang lebih besar. Bilangan random juga telah digunakan estetis, misalnya dalam sastra dan musik, dan tentu saja pernah populer untuk permainan dan perjudian. Ketika mendiskusikan angka tunggal, sebuah nomor acak yang diambil dari satu set nilai yang mungkin, masing-masing yang sama kemungkinan, yaitu, distribusi seragam. Ketika mendiskusikan urutan angka acak, setiap nomor diambil harus statistik independen dari yang lain.

Banyaknya aplikasi yang menggunakan bilang random sehingga menyebabkan pengembangan beberapa metode yang berbeda untuk menghasilkan data acak. Untuk memperoleh bilangan acak ini sudah ada sejak zaman dulu diantara melempar dadu, membalik koin , kartu , dan teknik lainnya. Karena sifat mekanik teknik ini, menghasilkan jumlah besar cukup nomor acak (penting dalam statistik) membutuhkan banyak pekerjaan dan/atau waktu. Dengan demikian, hasil kadang-kadang akan dikumpulkan dan didistribusikan sebagai tabel nomor acak.

Beberapa algoritma telah dikembangkan untuk membangkitkan bilangan random, salah satunya adalah dengan metode linear congruen. Metode linear congruent telah digunakan dalam berbagai aplikasi diantaranya digunakan untuk pengacak soal pada aplikasi tryout SMPTN[1], digunakan dalam algoritma kriptografi [2], selain itu juga digunakan pembuatan perangkat lunak penyimpanan data rahasia dengan menggunkan teknik stegonography untuk media citra digital [3].

Munthe menggunakan pada aplikasi tryout. Tryout merupakan kegiatan yang sangat lazim dilakukan oleh masyarakat apabila akan mengadakan ujian, terutama pada kalangan siswa-siswi yang akan menghadapi Seleksi Nasional Masuk Perguruan Tinggi Negeri. *Linear Congruent Method* (LCM) dapat digunakan sebagai dasar acuan pada berbagai aplikasi untuk menyelesaikan masalah yang mempunyai nilai acak dan *Linear Congruent Method* (LCM) juga dapat diimplementasikan pada Aplikasi *Tryout* SNMPTN. Pengacakan dengan *Linear Congruent Method* (LCM) menghasilkan nilai acak yang periodik, sehingga pemberian variabel yang selalu berubah-ubah dapat mengatasi keperiodikan nilai acak yang terjadi [1].

Sartono menggunakan pembangkitan bilangan acak untuk simulasi Monte Carlo banyak digunakan oleh para analis, misalnya oleh mereka yang menduga nilai resiko dari suatu portfolio bank [4].

Pada penelitian ini peneliti mencoba menguji kerandoman pembangkitan bilangan random dengan metode linear congruent, dan untuk pengujian kerandoman bilangan yang dihasilkan mengunakan uji chi-square.

## METODE PENELITIAN

*Random number* atau bilangan acak adalah sebuah bilangan yang dihasilkan dari sebuah proses, yang keluarannya tidak dapat diprediksi dan secara berurutan tidak bisa dihasilkan bilangan yang sama. Proses yang menghasilkan random number disebut *random number generator*. Walaupun kelihatannya cukup sederhana, dari definisinya, tetapi pada kenyataanya cukup sulit untuk menghasilkan bilangan yang benar-benar acak[5].

Saat ini, setelah munculnya nomor acak generator komputasi, semakin banyak, permainan lotore, menggunakan RNGs bukannya lebih metode menggambar tradisional, seperti menggunakan ping-pong atau bola karet. RNGs juga digunakan hari ini untuk menentukan kemungkinan mesin slot modern. [6] Beberapa metode komputasi untuk pembangkit nomor acak ada, tetapi sering jatuh pendek dari tujuan keacakan benar - meskipun mereka mungkin bertemu, dengan berbagai keberhasilan, beberapa uji statistik untuk keacakan dimaksudkan untuk mengukur seberapa hasil yang tak terduga mereka (yaitu, untuk apa pola derajat mereka yang jelas). *Testing randomness* bertujuan untuk menentukan apakah uratan beberapa bilangan dihasilkan oleh *random generator* atau bukan.

Random Number Generator adalah suatu algoritma yang digunakan untuk menghasilkan uruta-urutan angka angka random baik secara hitungan manual maupun komputasi elektronik (komputer). Bilangan acak disesuaikan dengan besar

probabilitas yaitu antara 0 s/d 1.0 dan berdistribusi seragam. Syarat Pembangkitan Bilangan acak antara lain bersifat random, tidak ber-ulang (Degenerative) dan perioda ulang biasanya munculnya sangat panjang

Ada dua pendekatan utama untuk menghasilkan nomor acak menggunakan komputer: True Random Number Generator (TRNGs) dan Pseudo-Random Number Generator (PRNGs). True Random Number Generator (TRNGs) adalah ekstrak keacakan dari fenomena fisik dan memperkenalkan ke dalam komputer. Anda bisa bayangkan ini sebagai mati terhubung ke komputer, tetapi biasanya orang menggunakan sebuah fenomena fisik yang lebih mudah untuk terhubung ke komputer daripada mati adalah. Fenomena fisik bisa sangat sederhana, seperti variasi kecil dalam gerakan mouse seseorang atau dalam jumlah waktu antara penekanan tombol. Dalam prakteknya, namun Anda harus berhati-hati tentang mana sumber yang Anda pilih. Sebagai contoh, bisa rumit untuk menggunakan penekanan tombol dengan cara ini, karena penekanan tombol sering buffered oleh sistem operasi komputer, yang berarti bahwa beberapa penekanan tombol dikumpulkan sebelum dikirim ke program menunggu mereka. Untuk sebuah program menunggu penekanan tombol, maka akan tampak seolah-olah tombol ditekan hampir bersamaan, dan mungkin tidak banyak keacakan sana setelah semua.

Karakteristik TRNGs umumnya lebih efisien dibandingkan dengan PRNGs, meluangkan waktu jauh lebih lama untuk menghasilkan angka. TRNGs juga nondeterministic, yang berarti bahwa urutan yang diberikan nomor tidak bisa direproduksi, meskipun urutan yang sama mungkin saja terjadi beberapa kali secara kebetulan.Perbandingan PRNGs dan TRNGs. Berikut ini beberapa pembangkit bilangan acak semu :

1. linear congruential generators (LCG)
2. Park-Miller random number generator
2. lagged Fibonacci generators
3. linear feedback shift registers
4. generalised feedback shift registers
5. Blum Blum Shub
6. Mersenne twister

Linear congruential method (LCM) adalah salah satu metode pembangkit bilangan random semu (Pseudorandom number generators). LCM dikenalkan oleh D.H. Lehmer pada tahun 1949 [7]. Metode ini merupakan pseudorandm number generator yang umum digunakan karena mudah diimplementasikan secara komputasi dan memiliki kecepatan yang relatif cepat [8]. Tabel 1 menunjukkan beberapa parameter yang umum digunakan untuk test pembangkitan bilangan random[4].

Tabel 1 Common Random-Number Generators

| Generator | A | M | C |
|---|---|---|---|
| RANF<br>CDC 60000 FTN compiler | 44485709377909 | $2^{48}$ | - |
| GGUBS<br>IMSL Routine | $7^5$ | $2^{31}$ | - |
| RANDU<br>IBM Scientific Subroutine | $2^{16} + 3$ | $2^{31}$ | - |
| CGL<br>IBM Subroutine Lib.-Math | $7^5$ | $2^{31}$ | - |

Validasi Bilangan Acak

Pengujian dimaksudkan untuk melihat distribusinya, urutan ke-acakan-nya.

Metoda pengujian :

1. Uji empiris
   dilakukan dengan uji statistik;
   - Chi-Square test : untuk menguji keseragaman
   - Run test : untuk menguji keacakan
2. Uji teoritis
   dilakukan uji parameter pembangkit untuk pembangkitan secara menyeluruh.
   - Spectral test
   - Lattice test

Dalam makalah ini akan membahas mengenai metode pembangkit bilang acak yaitu linear congruential generators (LCG) dilakukan pengujian validasi kerandoman dengan metode Chi-Square. Linear Congruent Method (LCM) merupakan metode pembangkitkan bilangan acak yang banyak digunakan dalam program komputer. Pembangkit bilangan acak kongruen-lanjar (*linear congruential generator* atau LCG) adalah *PRNG* yang berbentuk:

$$X_n = (aX_{n-1} + c) \bmod m \qquad (1)$$

Dimana :

$Xn$ = bilangan acak ke-$n$ dari deretnya
$X_{n-1}$ = bilangan acak sebelumnya
$a$ = faktor pengkali
$c$ = *increment*
$m$ = modulus

Kunci pembangkit adalah $X_0$ yang disebut umpan (*seed*).

Alasan di balik algoritma ini adalah pengamatan umum bahwa dalam perkalian dan pembagian, semakin rendah, agar angka lebih sulit untuk memprediksi. Menunjukkan tidak ada pola yang jelas atau urutan. Alasan yang sama digunakan oleh taruhan saat membuat jumlah loterry harian, mendasarkan pada beberapa digit terakhir dari total jumlah taruhan di trek balap tertentu. Jumlah itu tampaknya acak karena didasarkan pada angka yang paling tidak bisa diprediksi dan paling cepat berubah.

Contoh :

1. Membangkitkan bilangan acak sebanyak 8 kali dengan a = 2, c = 7, m = 10 dan x(0) = 2

$x(1) = (2(2) + 7) \bmod 10 = 1$
$x(2) = (2(1) + 7) \bmod 10 = 9$
$x(3) = (2(9) + 7) \bmod 10 = 5$
$x(4) = (2(5) + 7) \bmod 10 = 7$
$x(5) = (2(7) + 7) \bmod 10 = 1$
$x(6) = (2(1) + 7) \bmod 10 = 9$
$x(7) = (2(9) + 7) \bmod 10 = 5$
$x(8) = (2(5) + 7) \bmod 10 = 7$

Bilangan acak yang dibangkitkan adalah : 1 9 5 7 1 9 5 7
Terjadi pengulangan bilangan secara periodik(4)

2. Membangkitkan bilangan acak sebanyak 8 kali dengan a = 4, c = 7, m = 15 dan x(0) = 3

$x(1) = (4\ (3)\ + 7) \bmod 15 = 4$
$x(2) = (4\ (4)\ + 7) \bmod 15 = 8$
$x(3) = (4\ (8)\ + 7) \bmod 15 = 5$
$x(4) = (4\ (5)\ + 7) \bmod 15 = 12$
$x(5) = (4\ (12) + 7) \bmod 15 = 10$
$x(6) = (4\ (10) + 7) \bmod 15 = 2$
$x(7) = (4\ (2)\ + 7) \bmod 15 = 0$
$x(8) = (4\ (0)\ + 7) \bmod 15 = 7$

Bilangan acak yang dibangkitkan a dalah: 4 8 5 12 10 2 0 7
Tidak terlihat pengulangan bilangan secara periodik

3. Membangkitkan bilangan acak sebanyak 16 kali dengan a = 4, c = 7, m = 15 dan x(0) = 3

$x(1) = (4(3) + 7) \bmod 15 = 4$
$x(2) = (4(4) + 7) \bmod 15 = 8$
$x(3) = (4(8) + 7) \bmod 15 = 5$
$x(4) = (4(5) + 7) \bmod 15 = 12$
$x(5) = (4(12) + 7) \bmod 15 = 10$
$x(6) = (4(10) + 7) \bmod 15 = 2$
$x(7) = (4(2) + 7) \bmod 15 = 0$
$x(8) = (4(0) + 7) \bmod 15 = 7$
$x(9) = (4(7) + 7) \bmod 15 = 12$
$x(10) = (4(13) + 7) \bmod 15 = 14$
$x(11) = (4(14) + 7) \bmod 15 = 3$
$x(12) = (4(3) + 7) \bmod 15 = 4$
$x(13) = (4(4) + 7) \bmod 15 = 8$
$x(14) = (4(8) + 7) \bmod 15 = 5$
$x(15) = (4(5) + 7) \bmod 15 = 12$
$x(16) = (4(12) + 7) \bmod 15 = 10$

Terjadi pengulangan pada periode waktu tertentu atau setelah sekian kali pembangkitan, hal ini adalah salah satu sifat dari metode ini, dan pseudo random generator pada umumnya. Penentuan konstanta LCM (a, c dan m) sangat menentukan baik tidaknya bilangan acak yang diperoleh dalam arti memperoleh bilangan acak yang seakan-akan tidak terjadi pengulangan.

Metode Coungruantial telah terbukti untuk menghasilkan urutan nomor acak yang muncul untuk statistik acak, memberikan koefisien A dan C yang dipilih benar.

Uji Chi Square

Dibangkitkan 100 bilangan acak yang akan dikelompokkan dalam 10 kelompok kelas probabilitas.

Tabel 2 pembangkitan 100 Bilangan acak

| Kelas | Frekuensi Bilangan acak Fo | Frekuensi harapan Fe | (Fo-Fe)2/Fe Chi-sqre. |
|---|---|---|---|
| 0.0 – 0.09 | 9 | 10 | 0.1 |
| 0.1 – 0.19 | 12 | 10 | 0.4 |
| 0.2 – 0.29 | 10 | 10 | 0.0 |
| 0.3 – 0.39 | 11 | 10 | 0.1 |
| 0.4 – 0.49 | 8 | 10 | 0.4 |
| 0.5 – 0.59 | 10 | 10 | 0.0 |
| 0.6 – 0.69 | 10 | 10 | 0.0 |
| 0.7 – 0.79 | 7 | 10 | 0.9 |
| 0.8 – 0.89 | 12 | 10 | 0.4 |
| 0.9 – 1.00 | 11 | 10 | 0.1 |
| | | | 0.1 |
| | 100 | 100 | 2.4 |

- Pengujian:
  - Ho = data/acak terdistribusi seragam
  - H1 = Tidak terdistribusi seragam
  - Selang kepercayaan α = 0.05 (5%)
- Nilai Chi-square tabel = 16.919
  - Chi-square hitung = 2.4 artinya < nilai tabel
  - Kesimpulan terima Ho

Run Test

Mengujian dengan Urutan ke-acak-an diuji, dengan cara pengujian bilangan acak dalam urutannya bila harganya naik beri satu tanda +, sebaliknya tanda -, seterusnya sampai seluruh bilangan acak.

Contoh; 40 bilangan acak sbb;

0.43; 0.32; 0.48; 0.23; 0.90; 0.72; 0.94; 0.11; 0.14; 0.67;
0.61; 0.25; 0.45; 0.56; 0.87; 0.54; 0.01; 0.64; 0.65; 0.32;
0.03; 0.93; 0.08; 0.58; 0.41; 0.32; 0.03; 0.18; 0.90; 0.74;
0.32; 0.75; 0.42; 0.71; 0.66; 0.03; 0.44; 0.99; 0.40; 0.55.

Total run x = 26 (26 tanda + dan -)
Nilai harapan total run

$\mu = (2n-1)/3 = ((2x40)-1)/3 = 26.33$

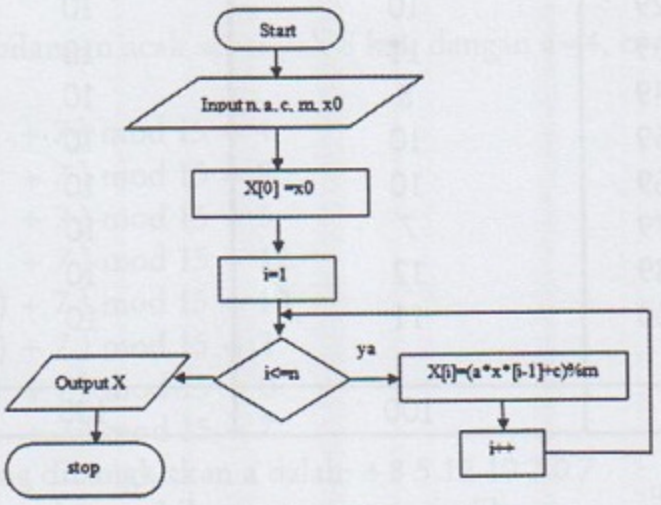
Variansi jumlah run: $\sigma 2 = (16n-29)/90 = ((16x40)-29)/90 = 6.79$
Standar deviasi $\sigma = 2.61$
Pengujian dengan distribusi normal;

Ho : $\mu = 26.33$, H1 = bukan $\mu$

$Z = (a-\mu)/\sigma = (26-26.33)/2.61 = -0.13$

Batas selang-kepercayaan -1.96 s/d 1.96., berarti harga Z ada didalamnya Terima Ho.

Langkah-langkah pembangkitan bilangan random dengan flowchart ditunjukkan pada gambar 1.




Gambar 1 Flowchart LCM

Dari Gambar 1 deklarasi variabel yang dibutuhkan dalam program adalah sebagai berikut

Deklarasi variabel :

n = bertipe integer berfungsi untuk menyimpan data jumlah bilangan acak yang akan dibangkitkan
a = bertipe integer berfungsi untuk menyimpan parameter a
c = bertipe integer berfungsi untuk menyimpan parameter c
m = bertipe integer berfungsi untuk menyimpan parameter m
X = bertipe array of integer berfungsi untuk menyimpan data hasil bilangan acak yang dibangkitkan
I = bertipe integer berfungsi sebagai counter increment pada saat pembangkitan bilangan acak

Implementasi

Aplikasi yang di buat adalah membuat fungsi untuk Pembangkin Bilangan Acak Semu dengan metode *linear congruential generator* kemudian hasil bilangan acak yang di generate di uji validasi Bilangan Acak untuk menguji keseragamannya dengan mengunakan uji Chi-Square. Gambar 2 Menunjukkan fungsi dari pembangkitan bilangan random dengan metode *linear congruential generator.*

```
public long[] LinearCongruentMethod(int n, int a, int c,
                                    int  m,  int x0){
    x[0] = x0;
    for (int i=1; i<=n; i++){
      x[i]=(a*x[i-1]+c) % m;
    }
    return x;
  }
```

Gambar 2 Fungsi dengan Metode LCM

Dari gambar 2 pemanggilang fungsi linnearCongruentMetode akan dihasilkan vektor berupa array bilangan random yang dihasilkan dari perhitungan dengan metode LCM. Fungsi terdiri dari 4 parameter yaitu n, a, d, m dan x0. N merupakan jumlah bilangan acak yang akan dibangkitkan yang nantinya digunakan sebagai pengujian kerandoman dengan uji chi-square. Variabel paramete a digunakan sebagai faktor pengkali, sedangkan c merupakan bilangan pembagi atau modulus. Pendefinisian fungsi bertujuan agar mudah dikembangkan dan digunakan kembali untuk aplikasi yang lain.

Hasil dari pembangkitan bilangan acak kemudian dilakukan proses pengujian kerandoman dengan chi-square. Fungsi untuk proses pengujian ditunjukkan pada Gambar 3.

```
public boolean chiTest(int n, i  nt k, double batas, int
m){
    int[] kelas=new int[200000];
    int[] frekAcak=new int[200000];
    int[] frekFe =new int[200000];
    double[] chi =new double[200000];
    int range = m/k;
    for (int i=0;i<k;i++){
        kelas[i] =i*range;
        frekFe[i] =range;
    }
    int j;
    for (int i=1;i<=n;i++){
        j=(int) x[i] / range;
        frekAcak[j]=frekAcak[j]+1;
    }
```

```
        for (int i=0;i<k;i++){
            chi[i]=Math.pow((frekAcak[i]-
                    frekFe[i]),2)/frekFe[i];
        }
        double sumChi=0;
        int sumFo=0, sumFe=0;

        for (int i=0;i<k;i++){
            sumChi=sumChi+chi[i];
            sumFo = sumFo + frekAcak[i];
            sumFe = sumFe + frekFe[i];
        }

        if (sumChi<batas)
          return true;
        else
          return false;
    }
```

Gambar 3 Fungsi uji Chi-square

Hasil Eksekusi program

Membangkitkan bilangan acak sebanyak 8 kali dengan a=2, c=7, m = 10 dan x(0)=2

```
Compiling 1 source file to
compile-single:
run-single:
1  9  5  7  1  9  5  7
Validasi Bilangan Acak (Chi-Square) =true
BUILD SUCCESSFUL (total time: 0 seconds)
```

Membangkitkan bilangan acak sebanyak n=8 kali dengan a=4, c=7, m = 15 dan x(0)=3

Output:

```
compile-single:
run-single:
4  8  9  13  14  3  4  8
Validasi Bilangan Acak (Chi-Square) =false
```

```
BUILD SUCCESSFUL (total time: 0 seconds)
```

Dari hasil pengujian menunjukakan adanya pengulangan pola bilangan acak yang dibangkitkan, hal ini telah disebutkan bahwa LCM merupakan pseudorandom number generator. Agar hasil dari bilangan acak yang dibangkitkan dapat memberi kepuasan pada pengguna maka perlu dilakukan menemukan nilai-nilai yang baik untuk parameter yang menentukan urutan congruential liner. Pertama mempertimbangkan pilihan yang tepat jumlah m. Alternatif lain adalah dengan membiarkan m menjadi bilangan prima terbesar kurang w.

Bilangan acak yang berkualitas baik antara bila terjadi perulangan atau munculnya bilangan acak yang sama setelah sekian periode tertentu (semakin lama semakin baik) dan bila terjadi perulangan kemunculannya tidak bisa diprediksi, yaitu pengaruhi oleh penentuan nilai awal Z0 dan konstanta (a, c, dan m) akan menentukan kualitas bilangan acak yang dihasilkan.

## KESIMPULAN

Algoritma pembangkitan bilangan acak dengan metode linear congruential generator menunjukkan bahwa bilangan yang dihasilkan adalah acak, namun berdasar pengujian chi-square penggunakaan metode linear congruential generator masih terdapat perulangan hal ini ditunjukkan bahwa validasi bilangan acak bernilai false. Penentuan konstanta LCM (a, c dan m) sangat menentukan baik tidaknya bilangan acak yang diperoleh.

## SARAN

Adapun saran yang dapat diberikan untuk penelitian selanjutnya yaitu pertama pembangkitan bilangan random dapat dibandingkan dengan metode pembangkitan lain yang setipe untup psedurandom number generator. Kedua metode pembangkitan bilangan random dapat digunakan pada aplikasi untuk pengacakan bilangan.

## DAFTAR PUSTAKA

[1] Munthe, D., 2014. Implementasi Linier Congruent Method (LCM) pada Aplikasi Tryout SNMPTN (Studi Kasus : Bimbingan Dan Pemantapan Belajar Quin Medan), Pelita Informatika Budi Darma, Volume : VII, Nomor: 2, Agustus 2014 ISSN : 2301-9425.

[2] Bellare, M., Goldwasser, S. and Micciancio, D., 1997. Pseudo Random Number Generation within Cryptographic Algorithms the DSS Case Cryptographic Algorithms : the DSS Case. *Appears in Advances in Cryptology – Crypto 97 Proceedings, Springer-Verlag.*

[3] Sulindawaty, 2011, Pembuatan Perangkat Lunak Penyimpan Data Rahasia dengan Menggunakan Teknik Steganography untuk Media Citra Digital, *Jurnal SAINTIKOM,* Vol. 10 / No. 3 / September 2011

[4] Sartono, B, 2005, Pembangkitan Bilangan Acak Untuk Simulasi Monte Carlo Non-Parametrik, Forum Statistika dan Komputasi, Oktoberl 2005, p: 8 - 11 Vol. 10 No. 2, ISSN : 0853-8115.

[5] Stewart V. Hoover, Ronald F. Perly. 1989. *SIMULATION A Problem Solving Approach*, Addison-Wesley Publishing Company, Chapter 7.

[6] Willian H.Press, Saul A. Teukolsky, William T. Vetterling, Brian P.Plannery. 2002, *Numerical Recipes in C++ the Art of Scientific Computing, Second Edition.* Combtridge University Press, chapter 7.

[7] D.H. Lehmer. Mathematical methods in large-scale computing units. In Proceedings of the Second Symposium on Large-Scale Digital Calculating Machinery, pages 141–146, Cambridge, MA, 1951. Harvard University Press.

[8] Glen. A., 2002. On the Period Length of Pseudorandom Number Sequences, *Thesis*, The University of Adelaide, Australia.