Semantika
Seminar Nasional Teknologi Informatika

Perbandingan Algoritma Pembangkit Bilangan *Linear Congruential Generator* (LCG) dengan *Multiplicative Random Number Generator* (MNRG) dalam Probabilitas Kemunculan Bilangan yang Sama

Naproni
Amik Akmi Baturaja
naproni@gmail.com

Abstrak

Pembangkit Bilangan Acak dibutuhkan berbagai bidang ilmu komputer seperti pada game, kriptografi, perancangan sistem yang membutuhkan kemunculan acak terhadap suatu bilangan dan lain lain. Pembangkit bilangan acak *Linear Congruential Generator* (LCG) dan *Multiplicative Random Number Generator* (MNRG) merupakan salah satu pembangkit bilangan acak yang mudah dipelajari. Pembangkit bilangan acak ini membutuhkan dua buah bilangan *primitive root* yang mudah dicari dengan menggunakan *Greatest Common Divisor* menggunakan Algoritma Euclid. , Dengan melalui percobaan bahwa dalam proses iterasi terjadi kemunculan bilangan yang sama secara periodik pada kedua algoritma tersebut. Untuk menghindari kemunculan bilangan yang sama secara periodik dapat dilakukan dengan syarat jumlah iterasi harus lebih kecil dari nilai m.

Keywords— Bilangan Acak; primitive root, Linear Congruential Generator, Multiplicative Number Random Generator.

1. PENDAHULUAN

Bilangan Acak adalah bilangan yang muncul secara sembarang. Namun Bilangan Acak bukanlah bilangan sembarangan, karena bilangan acak muncul berdasarkan rumus aritmatika.

Bilangan acak yang diproses secara algoritma dan komputer (pemrograman) bekerja berdasarkan rumus tertentu, sehingga hampir bisa dikatakan bahwa bilangan Acak bukanlah Acak. Bilangan yang diproses secara komputerisasi disebut dengan *Pseudo Random Number* (bilangan acak semu).

Kriteria yang harus dipenuhi oleh bilangan acak antara lain :

a) *Uniform:* Bilangan acak mempunyai distribusi yang sama.

b) Peluangan kemunculan angka berdasarkan range yang telah ditentukan harus sama besar. Tidak

Bilangan Acak banyak digunakan untuk permainan, yang semula dilakukan dengan melempar dadu atau dengan mengocok kartu. Dan

pada pembahasan berikut ini adalah membandingkan dua metode pembangkit bilangan acak yaitu *Linear Congruential Generator* dengan *Multiplicative Random Number Generator* membandingkan kemungkinan angka yang sama dengan melakukan berkali kali dalam hal ini dilakukan masing - masing 1000 kali percobaan.

2. LANDASAN TEORI

A. Operasi Modulus

Operasi Modulus atau Modulo (dalam bahasa pemrograman komputer disebut dengan mod) adalah operasi matematika yang menghasilkan sisa pembagian bilangan terhadap bilangan lainnya. Operasi bilangan ini digolongkan kepada Aritmatika.

Contoh operasi Modulo :

7 mod 3 = 1 → (7 dibagi 3 = 2 sisa 1)
9 mod 3 = 0 → (9 dibagi 3 = 3 sisa 0)
8 mod 5 = 3 → (8 dibagi 5 = 1 sisa 3)

B. Bilangan Relatif Prima

Dua buah bilangan disebut relatif prima jika faktor persekutuan terbesar (FPB) nya menghasilkan nilai 1. Bisa dikatakan bahwa kedua bilangan tersebut tidak memiliki faktor prima bersama.

Misalkan a = 10 dan b = 13. Maka bisa dikatakan a relatif prima terhadap b karena fpb(a,b) = 1. Bilangan a = 31 dan b = 36, maka a relatif prima terhadap b karena memang memiliki fpb(a,b) = 1.

Untuk menentukan dua buah bilangan relatif prima atau tidak bisa menggunakan faktor persekutuan terbesar menggunakan metode Euclidean.

C. Linear Congruential Generator (LCG)

Linear Congruential Generator (LRG) sebagai pembangkitan bilangan acak semua adalah dengan menggunakan rumus :

$$\mathbf{Z_n = (a * Z_{n-1} + c) \text{ modulo } m}$$

keterangan :

Z_n = Hasil bilangan acak iterasi ke n
a = Bilangan pengali
c = increment
m = operasi Modulo

Formula yang digunakan di atas dapat membangkitkan bilangan acak semu yang diproduksi secara iteratif dimana Z_0 digunakan sebagai *seed* atau umpan (*seed*) yang digunakan sebagai kunci awal membangkitkan bilangan acak.

Contoh hasil perhitungan dengan membangkitkan bilangan acak sebanyak 10 kali sebagai berikut :

Perhatikan contoh berikut ini :

Tabel Iterasi LCG

Iterasi	Z0	m	a	c	Zi
1	15	9	5	7	1
2	1	9	5	7	3
3	3	9	5	7	4
4	4	9	5	7	0
5	0	9	5	7	7
6	7	9	5	7	6

Berdasarkan tabel di atas, kemunculan bilangan acak dengan nilai Z_0 = 15, m = 9, a = 5 dan c = 7 maka di dapatkan bilangan acaknya 1, 3, 4, 0, 7, 6.

LCG memiliki syarat :

- **c** relatif prima terhadap **m**
- a > 0, m > 0

D. Multiplicative Random Number Generator (MRNG)

Multiplicative Random Number Generator menggunakan formula

$$\mathbf{Z_i = (a * Z_{i-1}) \text{ mod } m}$$

Rumus di atas menggunakan Z_0 sebagai pemicu atau disebut juga dengan benih (*seed*) untuk membangkitkan bilangan awal.

Keterangan rumus di atas :

Zi = Hasil bilangan acak ke – i
a = faktor pengali

c = *increment*
m = modulus
a = bilangan bulat positif dengan a ≥ 0
m = bilangan bulat positif dengan m ≥ 0

Catatan :
1. Semakin besar nilai m, maka semakin besar angka acak yang dihasilkan.
2. Nilai a dan m harus relatif prima.

Berikut ini hasil MRNG dengan nilai :

Z_0 = 2
a = 3
c = 5 dan
m = 1234

berdasarkan rumus di atas, maka didapatkan hasil sebagai berikut :

Z_1 = 11
Z_2 = 38
Z_3 = 119
Z_4 = 362
Z_5 = 1091
Z_6 = 810
Z_7 = 1201
Z_8 = 1140
Z_9 = 957
Z_{10} = 408

3. PEMBAHASAN

Pembahasan berikutnya adalah membahas tentang kemunculan bilangan yang sama pada range tertentu yang diambil pada jumlah data yang bilangan acak yang besar. Misalnya di range dalam 100 data, 1.000 data sampai dengan 1.000.000 perulangan sehingga membandingkan probabilitas kemunculan angka yang sama dalam periodik.

Pseudocode yang akan dilakukan adalah sebagai berikut :

a) Membangkitkan bilangan acak dengan range N bilangan
b) Menyimpan hasil kemunculan bilangan acak ke dalam list
c) Melakukan proses pengurutan data secara ascending
d) Menghitung frekwensi dan presentase kemunculan angka terbanyak dengan membandingkan jumlah angka yang dibangkitkan.
e) Melakukan hal yang sama pada metode LRG dan MNRG.
f) Mengambil kesimpulan metode terbaik antara LRG dan MNRG.

Berikut ini tabel yang dapat dilihat berdasarkan perbandingan antara LRG dan MNRG

Hasil Uji Coba LRG

1. Nilai z_0 = 2, a = 5, m = 7, c = 5 dengan iterasi 10 kali.

   hasil adalah : [**2, 1, 3, 6**, 0, 5, **2, 1, 3, 6**]
   Keterangan : terjadi perulangan periodik pada angka 2,1,3,6.

2. Nilai z_0 = 2, a = 90, m = 97, c = 5 dengan iterasi 100 Kali.
   Hasil adalah :
   [**2, 88, 68, 14,** 4, 74, 69, 7, 53, 22, 45, 78, 41, 9, 39, 23, 38, 30, 86, 82, 13, 11, 25, 24, 31, 79, 34, 58, 84, 96, 12, 18, 73, 76, 55, 8, 46, 71, 90, 54, 15, 94, 26, 17, 80, 27, 10, 32, 72, 83, 6, 60, 70, 0, 5, 67, 21, 52, 29, 93, 33, 65, 35, 51, 36, 44, 85, 89, 61, 63, 49, 50, 43, 92, 40, 16, 87, 75, 62, 56, 1, 95, 19, 66, 28, 3, 81, 20, 59, 77, 48, 57, 91, 47, 64, 42, **2, 88, 68, 14**]

   Terjadi kemunculan angka periodik (Perhatikan *bold*) pada angka **2, 88, 68, 14.**

3. Nilai z_0 = 2, a = 990, m = 997, c = 5 dengan iterasi 1000 Kali.
   Hasil adalah :
   [**2, 988, 68, 526**, 311, 819, 254, 221, 452, 829, 184, 711, 13, 911, 607, 741, 800, 387, 287, 987, 75, 477, 654, 412, 112, 218, 473, 682, 216, 487, 584, 902, 670, 300, 896, 712, 6, 960, 264, 151, 942, 390, 266, 137, 43, 701, 83, 421, 49, 659, 377, 357, 497, 514 ... ... 887, 775, 562, 59, 589, 867, 915, 579, 937, 425, 21, 855, **2, 988, 68, 526**]
   Juga terdapat perulangan periodik pada bagian awal dan bagian akhir data yaitu **2, 988, 68, 526**.

4. Nilai z_0 = 2, a = 992, m = 9973, c = 5 dengan iterasi 1000 Kali.

Hasil juga terjadi perulangan secara periodik pada bagian awal dan akhir dengan angka perulangan **2, 9956, 192, 7866, 3236, 4301, 2559, 1775, 426, 5292, 1631, 2010, 7814, 3808, 7982, 1960, 8364, 7731, 4721, 7912, 2730, 9867, 1171, 7070, 2019, 7715, 4897, 5976**

Hasil Uji Coba MNRG

Dengan menggunakan nilai variabel yang sama dengan hasil uji coba LRG, akan dicoba diaplikasikan pada **MNRG**

1. Uji coba 1 dengan Nilai $z_0 = 2$, a = 5, m = 7 dengan iterasi 10 kali. Dengan hasil : [**3, 1, 5, 4**, 6, 2, **3, 1, 5,4**]

2. Uji coba 2 dengan Nilai $z_0 = 2$, a = 90, m = 97 dengan iterasi 100 kali.

   Hasil bilangan acak = [**83, 1, 90, 49**, 45, 73, 71, 85, 84, 91, 42, 94, 21, 47, 59, 72, 78, 36, 39, 18, 68, 9, 34, 53, 17, 75, 57, 86, 77, 43, 87, 70, 92, 35, 46, 66, 23, 33, 60, 65, 30, 81, 15, 89, 56, 93, 28, 95, 14, 96, 7, 48, 52, 24, 26, 12, 13, 6, 55, 3, 76, 50, 38, 25, 19, 61, 58, 79, 29, 88, 63, 44, 80, 22, 40, 11, 20, 54, 10, 27, 5, 62, 51, 31, 74, 64, 37, 32, 67, 16, 82, 8, 41, 4, 69, 2, **83, 1, 90, 49**]

   Terdapat perulangan periodik pada bagian awal iterasi dan akhir iterasi yaitu **83, 1, 90, 49**

3. Uji coba 3 dengan Nilai $z_0 = 2$, a = 652, m = 997 dengan iterasi 1000 kali.

4. Hasil uji coba juga masih terdapat perulangan pada bagian awal iterasi dan akhir iterasi. Hasil sebagai berikut : [**307, 764, 625, 724**, 467, 399, 928, 874, 561, 870, 944, 339, 691, 885, 754, 87, 192 ...., 986, 804, 783, 52, 6, 921, 298, 878, 178, 404, 200, 790, 628, 686, 616, 838, 20, 79, 661, 268, 261, 682, 2, **307, 764, 625, 724**]

4. KESIMPULAN

Setelah melakukan uji coba terhadap kedua metode LRG dan MNRG, dapat diambil kesimpulan sebagai berikut :

1. Kedua metode tetap mengeluarkan perulangan bilangan secara iteratif
2. Tingkat banyaknya perulangan kemunculan bilangan yang sama pada proses tersebut sangat tergantung pada nilai ***m*** dan nilai ***a***.
3. Dengan terjadinya perulangan kemunculan bilangan yang sama secara periodik maka bisa disimpulkan bahwa pembangkit bilangan acak ini tidaklah benar benar acak / atau tidak sepenuhnya acak.
4. Untuk menghindari kemunculan bilangan acak secara periodik dapat dilakukan dengan syarat jumlah iterasi < *m*. Dengan kondisi ini maka tidak terjadi kemunculan bilangan yang sama secara periodik
5. Agar terlihat benar benar acak (*menuju kesempurnaan*) perlu membangkitkan bilangan yang acak untuk Umpan (Z_0) agar tidak menghasilkan nilai konstan

5. DAFTAR PUSTAKA

[1] Andika, Yohanes, and Rinaldi Munir. "Pengembangan Random Number Generator dengan Video dan Suara." Jurnal Sarjana ITB bidang Teknik Elektro dan Informatika 1.2 (2012).

[2] Mohamad Octamanullah, “Analisa Mendalam Pustaka Pembangkit Bilangan Acak Semu Pada Linux”, ITB, 2007

[3] Ramadhan, Andresta. "Perbandingan Algoritma Linear Congruential Generators, BlumBlumShub, dan MersenneTwister untuk Membangkitkan Bilangan Acak Semu." Institut Teknologi Bandung. Bandung (2011).

[4] Sartono, Bagusi. "Pembangkitan Bilangan Acak Untuk Simulasi Monte Carlo Non-Parametrik." FORUM STATISTIKA DAN KOMPUTASI. Vol. 10. No. 2. 2005.