

MODIFIKASI *LINEAR CONGRUENTIAL GENERATOR* UNTUK OPTIMALISASI PENILAIAN PEMBELAJARAN *COMPUTER BASED TEST* (CBT)

Hendro Eko Prabowo*, Arimaz Hangga

Jurusan Teknik Elektro, Fakultas Teknik, Universitas Negeri Semarang
Kampus UNNES, Kel. Sekaran, Kec. Gunungpati, Semarang, Jawa Tengah, 50229.

*Email: hendro.prabowo15@gmail.com

Abstrak

Model penilaian pembelajaran konvensional menggunakan kertas atau Paper Based Test (PBT) memiliki kekurangan yaitu rentan terhadap kecurangan yang disebabkan oleh kesamaan soal untuk masing-masing peserta tes. Disisi lain PBT memiliki keterbatasan dalam hal sumber daya yang digunakan misal sumber daya manusia, waktu, dan biaya. Salah satu cara untuk mengurangi keterbatasan PBT adalah mengembangkan model penilaian pembelajaran berbasis komputer atau Computer Based Test (CBT). Pencegahan kesamaan soal untuk masing-masing peserta tes juga dilakukan dengan menerapkan sistem pengacakan soal pada CBT. Algoritma random number generator seperti Modifikasi Linear Congruential Generator (MLCG) dapat digunakan untuk membangun sistem tersebut. Penggunaan variabel yang berbeda pada parameter MLCG juga memberikan hasil yang berbeda dalam hal persentase kesamaan soal. Penggunaan bilangan bersifat koprima dan fibonacci akan memiliki persentase kesamaan soal 20%-35%; koprima dan prima 10%-35% dan bilangan dengan sifat ketiganya 10%-35%.

Kata kunci: *Computer Based Test, Modifikasi Linear Congruential Generator, Pengacakan*

1. PENDAHULUAN

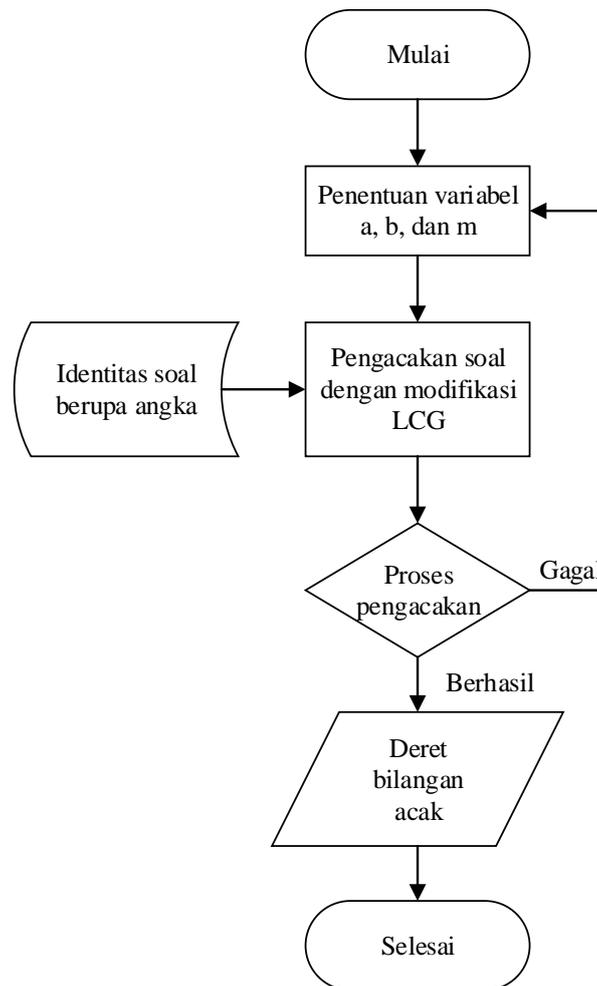
Konsep penilaian *Computer Based Test* (CBT) mulai dikembangkan dan diterapkan dalam proses penilaian hasil belajar siswa. Penilaian CBT memiliki kelebihan antara lain : mengurangi sumber daya manusia, lebih efisien dalam penggunaan waktu dan biaya pelaksanaan tes dibandingkan dengan *Paper Based Test* (PBT) (Simin, 2013). Sistem CBT dapat ditingkatkan dengan menerapkan sistem pengacakan soal untuk mengurangi kecurangan selama proses tes berlangsung seperti yang terjadi pada PBT. Salah satu sebab kecurangan dapat terjadi dikarenakan adanya kesamaan soal pada masing-masing peserta tes (McCabe, 1994).

Sistem pengacakan soal ini dapat dibangun dengan memanfaatkan metode *Random Number Generator* (RNG). Metode RNG dapat dibedakan menjadi dua jenis yaitu *True Random Number Generator* (TRNG) dan *Pseudo Random Number Generator* (PRNG). Penggunaan metode TRNG dapat digunakan sebagai pembangkit bilangan acak nyata yang dibentuk melalui sumber bilangan yang telah diacak (Stallings, 2011). Berbeda dengan metode PRNG yang menghasilkan bilangan acak melalui algoritma kriptografi dengan sumber pembangkit dari variabel-variabel yang digunakan pada algoritma (Martin, 2012). *Linear Congruential Generator* (LCG) merupakan contoh algoritma PRNG yang bersifat linear. Metode LCG memanfaatkan algoritma rekursif linear yang dikombinasi dengan fungsi modulus (Salomon, 2003). Adanya kebutuhan untuk mengurangi kecurangan pada CBT, maka penelitian ini diharapkan dapat memberikan solusi dengan sistem pengacakan soal menggunakan *Modifikasi Linear Congruential Generator* (MLCG).

2. METODE PENELITIAN

Pembentukan sistem pengacakan soal dimulai dengan menentukan nilai parameter yang akan digunakan pada algoritma MLCG. Parameter yang harus ditentukan adalah parameter a , b dan m . Variabel yang digunakan pada parameter m dibuat bervariasi sehingga akan diketahui tingkat perbedaan pola yang dihasilkan oleh metode modifikasi MLCG. Variasi yang dimaksud adalah membuat variabel m menjadi beranekaragam yaitu 10, 20, 30, dan 40 dimana nilai tersebut mewakili jumlah soal yang digunakan pada tes CBT. Sedangkan untuk parameter b akan digunakan bilangan prima, koprima dan fibonacci sebagai pembatasan jumlah variabel dalam penelitian ini.

Hasil dari penelitian ini akan menunjukkan persentase kesamaan soal berdasarkan pemilihan bilangan yang digunakan pada parameter b .



Gambar 1. Algoritma pembangunan sistem pengacakan soal

Terlihat pada **Gambar 1**, bahwa setelah penentuan identitas soal yang akan digunakan pada tes model CBT berhasil didapatkan maka mulai dilakukan pengacakan soal. Proses ini menggunakan algoritma MLCG yang telah disesuaikan menjadi bentuk matrik. Apabila terjadi kegagalan maka sistem akan meminta penentuan ulang variabel a , b dan m . Hasil dari pengacakan adalah deret bilangan yang merupakan identitas masing-masing soal.

3. HASIL PENELITIAN DAN PEMBAHASAN

3.1 Pembentukan Model Matematis

Modifikasi *Linear Congruential Generator* (MLCG) yang digunakan sebagai model matematis dalam pengacakan soal diturunkan dari algoritma *Linear Congruential Generator* (LCG). Persamaan LCG digunakan untuk menentukan deret angka dimana deret tersebut akan dimasukkan kedalam matrik pada persamaan MLCG. Model matematis algoritma LCG dapat dilihat sebagai berikut (Knuth, 1981):

$$x_{n+1} = ((a \times x_n) + b) \bmod m \quad (1)$$

dengan x_{n+1} adalah bilangan acak ke- n dari deretnya, a adalah parameter pengali, x_n adalah bilangan acak sebelumnya, b adalah parameter penambah dan m adalah parameter modulus. Persamaan tersebut akan diturunkan menjadi bentuk matrik dengan orde $x \times y$. Model matematis yang digunakan untuk menentukan orde x pada baris matrik sebagai berikut (Katti, 2010):

$$x_{n+1} = ((a_1 \times x_n) + b_1) \text{ mod } m \quad (2)$$

sedangkan model matematis untuk menentukan orde y sebagai kolom matrik yaitu :

$$y_{n+1} = ((a_2 \times y_n) + b_2) \text{ mod } m \quad (3)$$

dengan x_{n+1} sebagai bilangan acak orde x ke- n dari deretnya, y_{n+1} sebagai bilangan acak orde y ke- n dari deretnya, x_n sebagai bilangan acak sebelumnya pada orde x , dan y_n sebagai bilangan acak sebelumnya pada orde y .

Model matematis yang telah diturunkan dari LCG kemudian disesuaikan pada matrik yang digunakan. Apabila matrik yang digunakan pada sistem pengacakan soal ini disebut sebagai matrik M , maka pembentukan matrik M dapat menggunakan persamaan berikut (Biantara, 2015):

$$M_n = M[x_{n+1} \text{ mod } i][y_{n+1} \text{ mod } j] \quad (4)$$

dimana i adalah jumlah baris matrik dan j adalah jumlah kolom pada matrik sedangkan n merupakan bilangan 0, 1, 2, 3, dan seterusnya.

3.2 Persentase Perbedaan Variabel m

Perbedaan penggunaan parameter m yang mewakili jumlah soal tes dan pembatasan penggunaan variabel pada parameter b berhasil dilakukan dalam MLCG. Terlihat pada **Tabel 1**. bahwa persentase kesamaan soal pada algoritma LCG umum berkisar antara 17%-24%. Sedangkan untuk algoritma MLCG mempunyai persentase kesamaan soal berkisar 20%-35% untuk penggunaan bilangan koprima dan fibonacci, 10%-35% untuk bilangan koprima dan prima, dan 10%-35% untuk ketiga bilangan tersebut.

Tabel 1. Persentase Kesamaan Soal

Jumlah Soal	Rata-rata Persentase Kesamaan Soal			
	LCG	Koprima dan Fibonacci	Koprima dan Prima	Koprima, Prima dan Fibonacci
10	20%	20,0%	10,0%	10,0%
20	24%	35,0%	35,0%	35,0%
30	19%	33,0%	26,5%	26,5%
40	17%	22,5%	22,5%	22,5%

Berdasarkan **Tabel 1**. terlihat bahwa persentase kesamaan soal tidak berubah secara signifikan apabila dilakukan pembatasan penggunaan bilangan pada variabel b dibandingkan dengan algoritma LCG. Hal ini dikarenakan adanya pembatasan pada variabel b akan memberikan dampak keterbatasan pola distribusi soal. Disisi lain persentase kesamaan soal juga dipengaruhi oleh penggunaan bilangan acak awal yang sama. Jika terjadi periode penuh pada pembentukan deret acak akan terdapat pengulangan bilangan pada posisi yang sama.

4. KESIMPULAN

Penggunaan algoritma MLCG mempunyai persentase kesamaan soal berkisar 20%-35% untuk penggunaan bilangan koprima dan fibonacci, 10%-35% untuk bilangan koprima dan prima,

dan 10%-35% untuk ketiga bilangan tersebut. Penggunaan pembatasan bilangan pada variabel b tidak memberikan perubahan yang signifikan dalam hal persentase kesamaan soal dibandingkan dengan algoritma LCG.

DAFTAR PUSTAKA

- Biantara, I Made D. dkk., (2015), Implementation Coupled Linear Congruential Generator Methods for Questions of Pattern Randomization, *International Conference on Vocational Education and Electrical Engineering (ICVEE)*, pp. 247-250.
- Katti, R. S. dkk., (2010), Pseudorandom Bit Generation Using Coupled Congruential Generator, *IEEE Transactions on Circuits and System II: Express Briefs Vol.57*, pp. 203-207.
- Knuth, D. E., (1981), *The Art of Computer Programming*, 2nd Ed., Addison-Wesley Publishing Company, Canada, pp. 9-16.
- Martin, Keith M, (2012), *Everyday Cryptography: Fundamental Principles and Applications*, Oxford University Press Inc., New York, pp. 253-259.
- McCabe, D. L. dan W. J. Bowers, (1994), Academic Dishonesty Among Males in College: A Thirty Year Perspective, *Journal of College Student Development Vol. 5*, pp. 5-10.
- Salomon, David, (2003), *Data Privacy and Security*, Springer-Verlag New York Inc., New York, pp. 97-100.
- Simin, Shahla dan A. Heidari,(2013), Computer-based Assessment: Pros and Cons,*Elixir International Journal Vol. 55*, pp. 12732-12734.
- Stalling, W., (2012),*Cryptography and Network Security: Principles and Practice*, 5thEd., Prentice Hall, New York, pp. 218-228.