

Aplikasi Pembangkit Bilangan Acak Berbasis Kekongruenan Lanjar Pada Gim Sudoku

Kartini Copa - 13521026
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13521026@itb.ac.id

Abstract— Sudoku merupakan salah satu permainan teka-teki logika penempatan angka kombinatorial. Sudoku erat kaitannya dengan konsep teori bilangan. Dari konsep teori bilangan, dapat diturunkan konsep pembangkit bilangan acak berbasis kekongruenan lanjar atau Linear Congruential Generator (LCG). Pada makalah ini akan dibahas mengenai berbagai cara mengimplementasikan konsep pembangkit bilangan acak berbasis kekongruenan lanjar pada Sudoku.

Keywords— Bilangan, LCG, Sudoku, Teori Bilangan.

I. PENDAHULUAN

Sudoku merupakan permainan legendaris yang mengasah logika penempatan angka. Seiring perkembangan teknologi, permainan Sudoku semakin bervariasi. Pada makalah ini, penulis berfokus pada Sudoku klasik. Sudoku klasik terdiri dari sembilan blok 3x3. Cara bermain Sudoku adalah dengan mengisi angka dari 1-9 pada setiap kotak yang kosong dengan aturan pada setiap baris dan kolom tidak ada angka yang berulang.

3			6	5				
1			7	2				
		5	4	1		2	3	
		3						
9			1	8	4	7		
	4			7	3			
3	4							
			1	5	2			

Pengacakan soal pada permainan Sudoku merupakan aplikasi dari konsep teori bilangan. Teori bilangan adalah cabang dari matematika murni yang mempelajari bilangan bulat dan sifatnya. Bilangan acak Sudoku merupakan aplikasi dari konsep teori bilangan yaitu pembangkit bilangan acak atau *pseudo random number generator* (PRNG). *Linear congruential generator* (LCG) mewakili salah satu algoritma PRNG yang tertua dan terpopuler untuk membangkitkan bilangan acak. PRG dan LCG merupakan aplikasi teori bilangan. Keuntungan dari LCG adalah operasinya yang sangat cepat. LCG dapat diterapkan untuk menghasilkan sekumpulan

nilai acak atau dapat digunakan untuk mengacak posisi dari sekumpulan nilai. Berdasarkan uraian di atas, penulis tertarik untuk menganalisis penerapan linear congruential generator (LCG) pada Sudoku.

II. LANDASAN TEORI

A. Teori Bilangan

Teori bilangan merupakan cabang ilmu matematika murni yang ditujukan untuk mempelajari *integer* atau bilangan bulat atau fungsi bernilai bilangan bulat. Sifat pembagian bilangan bulat: a habis membagi b jika terdapat bilangan bulat c sedemikian sehingga $b = ac$.

Teorema 1 (Teorema Euclidean). Terdapat bilangan bulat m dan n , dengan $n > 0$. Jika m dibagi n maka hasil pembagiannya adalah q (*quotient*) dan sisanya r (*remainder*), sehingga $m = nq + r$.

Teorema 2. Terdapat bilangan bulat m dan n , dengan $n > 0$, sehingga $m = nq + r, 0 \leq r < n$. Maka pembagi bersama terbesar (PBB - *greatest common divisor*) dari (m, n) sama dengan PBB dari (n, r) .

Teorema 3. Terdapat bilangan bulat positif a dan b , maka terdapat bilangan bulat m dan n sehingga $\text{PBB}(a, b) = ma + nb$.

Teorema 4. Terdapat bilangan bulat positif m .

1. Jika $a \equiv b \pmod{m}$ dan c adalah sembarang bilangan bulat maka:
 - i. $(a + c) \equiv (b + c) \pmod{m}$
 - ii. $ac \equiv bc \pmod{m}$
 - iii. $a^p \equiv b^p \pmod{m}$
2. Jika $a \equiv b \pmod{m}$ dan $c \equiv d \pmod{m}$, maka:
 - i. $(a + c) \equiv (b + d) \pmod{m}$
 - ii. $ac \equiv bd \pmod{m}$

B. Bilangan Acak

Bilangan dinyatakan sebagai bilangan acak apabila kemunculannya hanya dapat ditebak dengan kemungkinan. Bilangan acak adalah bilangan yang dihasilkan secara acak. Bilangan acak yang dihasilkan dari formula matematika adalah bilangan acak semu karena pembangkitannya dapat diulang kembali. Pembangkit deret bilangan acak semacam itu disebut *pseudo random number* (PRN).

C. Pembangkit Bilangan Acak

Struktur pembangkitan bilangan acak semu terdiri atas tiga bagian utama. Pertama, jumlah informasi pada pesan sistem operasi mulai dikumpulkan dari berbagai kejadian yang ditangkap oleh kernel pada komputer. Kedua, informasi pesan yang berhasil dikumpulkan diberikan kepada sebuah entitas penampung yang akan melakukan pencampuran dan penanganan kejadian yang diberikan oleh sistem operasi. Bagian terakhir terjadi jika bilangan acak tersebut dibutuhkan oleh pengguna. Bilangan acak diberikan kepada pengguna melalui beberapa algoritma pembangkit bilangan. Setelah bilangan acak diberikan, dimulai lagi tahap pertama. Algoritma pembangkit bilangan acak adalah algoritma dengan masukan minimum dapat menghasilkan suatu deret bilangan acak. Bila masukannya sama, maka akan menghasilkan urutan bilangan acak yang sama.

Pembangkit bilangan acak atau *random number generator* (RNG) merupakan suatu proses yang menghasilkan serangkaian bilangan atau simbol yang urutannya sulit diprediksi sehingga terlihat acak. Suatu pembangkit bilangan acak idealnya dapat menghasilkan bilangan yang tidak dapat diprediksi. Terdapat dua jenis pembangkit bilangan acak. Pertama, perangkat keras pembangkit bilangan acak atau *hardware random number generator* (HRNG) yang menghasilkan bilangan-bilangan acak sebagai fungsi dari nilai terkini dari suatu keadaan fisik yang selalu berubah sedemikian sehingga tidak bisa dimodelkan. Kedua, pembangkit bilangan acak semu atau *pseudo random number generator* (PRNG) yang menghasilkan bilangan-bilangan yang tampak acak, tetapi deterministik. Sehingga, pembangkit bilangan acak semu dapat dihasilkan kembali apabila keadaan-keadaan tersebut diketahui. Pembangkit bilangan acak semu juga dikenal sebagai *deterministic random bit generator* (DRBG). Pembangkit bilangan acak semu merupakan algoritma yang digunakan untuk membuat urutan dari himpunan bilangan sehingga urutan dari himpunan bilangan tersebut teracak. Suatu pembangkit bilangan acak semu dapat dimanipulasi urutannya dengan mengatur frekuensi. Terdapat berbagai macam pembangkit bilangan acak semu, salah satunya pembangkit bilangan acak berbasis kekongruenan lanjar yang memanfaatkan konsep teori bilangan.

C. Linier Congruential Generator

Sistem komputer menggunakan operasi aritmatika atau algoritma untuk membangkitkan bilangan acak. Tidak ada komputasi yang benar-benar menghasilkan deret bilangan acak secara sempurna. Bilangan acak yang dihasilkan dengan rumus-rumus matematika adalah bilangan acak semu (pseudo), karena pembangkitan bilangannya dapat diulang kembali. Terdapat banyak algoritma atau metode yang dapat digunakan. *pseudo random number generator* (PNRG) merupakan sebuah algoritma yang membangkitkan deret bilangan yang tidak benar-benar acak. Keluaran dari PNRG hanya mendekati beberapa sifat bilangan acak. *Linear congruential generator* (LCG) merupakan pembangkit bilangan acak berbasis kekongruenan lanjar. LCG juga merupakan metode pembangkit bilangan acak yang

banyak digunakan dalam program komputer. Keunggulan LCG adalah penggunaannya efisien karena cepat dan hanya membutuhkan sedikit operasi bit.

Linier congruential generator (LCG) memanfaatkan model linier untuk membangkitkan bilangan acak yang didefinisikan sebagai berikut,

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

dengan X_n adalah bilangan acak ke- n , X_{n-1} adalah bilangan acak sebelumnya, a adalah faktor pengali, b adalah *increment*, dan m adalah modulus. X_0 adalah kunci pembangkit yang disebut *seed* atau bilangan awal.

D. Algoritma Linier Congruential Generator

Kualitas pembangkitan bilangan acak dari setiap algoritma *pseudo random number generator* (PRNG) berbeda-beda. Pendekatan yang dibahas pada makalah ini adalah pembangkitan bilangan acak dengan menggunakan sebuah algoritma yang bisa membangkitkan sekuens bilangan dari sebuah *seed* atau bilangan awal. Berikut merupakan algoritma pembangkit bilangan acak berbasis kekongruenan lanjar dalam bentuk notasi algoritma.

```
function isMember (L: array of integer,
                  x: integer) → boolean
{ Mengembalikan true jika x merupakan member
  dari L dan mengembalikan false jika bukan
  member dari L }

procedure add (input/output x: integer,
              L: array of integer)
{I.S. L terdefinisi dan mungkin kosong}
{F.S. x ditambahkan sebagai elemen terakhir
  pada L}

procedure LCG (input a, b, m, seed:integer,
               output num: array of integer)
{I.S. b, m, dan seed terdefinisi}
{F.S. num bilangan-bilangan bulat yang
  dihasilkan oleh LCG}

KAMUS
  init: integer

ALGORITMA
  init ← seed
  while (not isMember(num, init)) do
    add(init, num)
    num ← (a * num + b) mod m
```

III. DESKRIPSI MASALAH

Terdapat tiga standar yang harus dipenuhi dalam pembuatan Sudoku. Pertama, teka-teki angka Sudoku harus memiliki solusi. Artinya, setiap kotak dalam baris dan kolom dipastikan dapat diisi angka 1-9 hanya sekali. Jadi, angka hanya dapat muncul sekali dalam setiap baris, kolom, dan blok. Kedua, terdapat penyelesaian logika untuk semua teka-teki Sudoku. Standar yang ketiga adalah standar estetika Sudoku, yaitu terkait susunan simetri dan distribusi normal.

Rancangan awal pembuatan soal acak Sudoku adalah menyediakan embilan blok 3x3. Pada permainan Sudoku terdapat tiga level atau tingkatan permainan yaitu, level mudah, level medium, dan level sulit.

IV. ANALISIS DAN IMPLEMENTASI

A. Alur Program

Berikut merupakan alur pembangkitan bilangan acak pada Sudoku di setiap blok-bloknya.

	8	3		7				
9				4				
2		4				7		8
3			4	6		5	1	
				3	2		4	
8		6	7		5	3	9	2
	1	8		9	7			3
5	3		6		4		8	
4	2		3	8				5

Pada Sudoku di atas, permainan Sudoku terdiri atas 39 bilangan acak. Membangkitkan bilangan acak sebanyak 39 kali yang dibagi dalam sembilan blok. Setiap blok minimal terdiri dari 2 bilangan acak dan maksimal 6 bilangan acak sebagai berikut.

- Blok 1

Sembilan kotak terdiri atas 5 bilangan acak angka 1-9 yang berbeda.

Z_2	8
Z_3	3
Z_4	9
Z_7	2
Z_9	4

- Blok 2

Sembilan kotak terdiri atas 2 bilangan acak angka 1-9 yang berbeda. Z_2 pada blok 2 tidak boleh sama dengan bilangan acak pada setiap baris kesatu dan setiap kolom kelima. Z_5 pada blok 2 tidak boleh sama dengan bilangan acak pada setiap baris kedua dan setiap kolom kelima.

Z_2	7
Z_5	4

- Blok 3

Sembilan kotak terdiri atas 2 bilangan acak angka 1-9 yang berbeda. Z_7 pada blok 2 tidak boleh sama dengan bilangan acak pada setiap baris ketiga dan setiap kolom ketujuh. Z_9 pada blok 3 tidak boleh sama dengan bilangan acak pada setiap baris ketiga dan setiap kolom kesembilan.

Z_7	7
Z_9	8

- Blok 4

Sembilan kotak terdiri atas 3 bilangan acak angka 1-9 yang berbeda. Z_1 pada blok 4 tidak boleh sama dengan bilangan acak pada setiap baris keempat dan setiap kolom kesatu. Z_7 pada blok 4 tidak boleh sama dengan bilangan acak pada setiap baris keenam dan setiap kolom kesatu. Z_9 tidak boleh sama dengan bilangan acak pada setiap baris keenam dan setiap kolom ketiga.

Z_1	3
Z_7	8
Z_9	6

- Blok 5

Sembilan kotak terdiri atas 6 bilangan acak angka 1-9 yang berbeda. Z_1 pada blok 5 tidak boleh sama dengan bilangan acak pada setiap baris keempat dan setiap kolom keempat. Z_2 pada blok 5 tidak boleh sama dengan bilangan acak pada setiap baris keempat dan setiap kolom kelima. Z_5 pada blok 5 tidak boleh sama dengan bilangan acak pada setiap baris kelima dan kolom kelima. Z_6 pada blok 5 tidak boleh sama dengan bilangan acak pada setiap baris kelima dan setiap kolom keenam. Z_7 pada blok 5 tidak boleh sama dengan bilangan acak pada setiap baris keenam dan setiap kolom keenam. Z_9 pada blok 5 tidak boleh sama dengan bilangan acak pada setiap baris keenam dan setiap kolom keenam.

Z_1	4
Z_2	6
Z_5	3
Z_6	2
Z_7	7
Z_9	5

- Blok 6

Sembilan kotak terdiri atas 6 bilangan acak angka 1-9 yang berbeda. Z_1 pada blok 6 tidak boleh sama dengan bilangan acak pada setiap baris keempat dan setiap kolom ketujuh. Z_2 pada blok 6 tidak boleh sama dengan bilangan acak pada setiap baris keempat dan setiap kolom kedelapan. Z_5 pada blok 6 tidak boleh sama dengan bilangan acak pada setiap baris kelima dan setiap kolom kedelapan. Z_7 pada blok 6 tidak boleh sama dengan bilangan acak pada setiap baris keenam dan setiap kolom ketujuh. Z_8 pada blok 6 tidak boleh sama dengan

bilangan acak pada setiap baris keenam dan setiap kolom kedelapan. Z_9 pada blok 6 tidak boleh sama dengan setiap baris kolom keenam dan setiap kolom kesembilan.

Z_1	5
Z_2	1
Z_5	4
Z_7	3
Z_8	9
Z_9	2

- Blok 7

Sembilan kotak terdiri atas 6 bilangan acak angka 1-9 yang berbeda. Z_2 pada blok 7 tidak boleh sama dengan bilangan acak pada setiap baris ketujuh dan setiap kolom kedua. Z_3 pada blok 7 tidak boleh sama dengan bilangan acak pada setiap baris ketujuh dan setiap kolom ketiga. Z_4 pada blok 7 tidak boleh sama dengan bilangan acak pada setiap baris kedelapan dan setiap kolom kesatu. Z_5 pada blok 7 tidak boleh sama dengan bilangan acak pada setiap baris kedelapan dan setiap kolom kedua. Z_7 pada blok 7 tidak boleh sama dengan bilangan acak pada setiap baris kesembilan dan setiap kolom kesatu. Z_8 pada blok 7 tidak boleh sama dengan bilangan acak pada setiap baris ke sembilan dan setiap kolom kedua.

Z_2	1
Z_3	8
Z_4	5
Z_5	3
Z_7	4
Z_8	2

- Blok 8

Sembilan kotak terdiri atas 6 bilangan acak angka 1-9 yang berbeda. Z_2 pada blok 8 tidak boleh sama dengan bilangan acak pada setiap baris ketujuh dan setiap kolom kelima. Z_3 pada blok 8 tidak boleh sama dengan bilangan acak pada setiap baris ketujuh dan setiap kolom keenam. Z_4 pada blok 8 tidak boleh sama dengan bilangan acak pada setiap baris kedelapan dan setiap kolom keempat. Z_6 pada blok 8 tidak boleh sama dengan bilangan acak pada setiap baris kedelapan dan setiap kolom keempat. Z_7 pada blok 8 tidak boleh sama dengan bilangan acak pada setiap baris kesembilan dan setiap kolom keempat. Z_8 pada blok 7 tidak boleh sama dengan bilangan acak pada setiap baris ke sembilan dan setiap kolom kelima.

Z_2	9
Z_3	7
Z_4	6
Z_6	4
Z_7	3
Z_8	8

- Blok 9

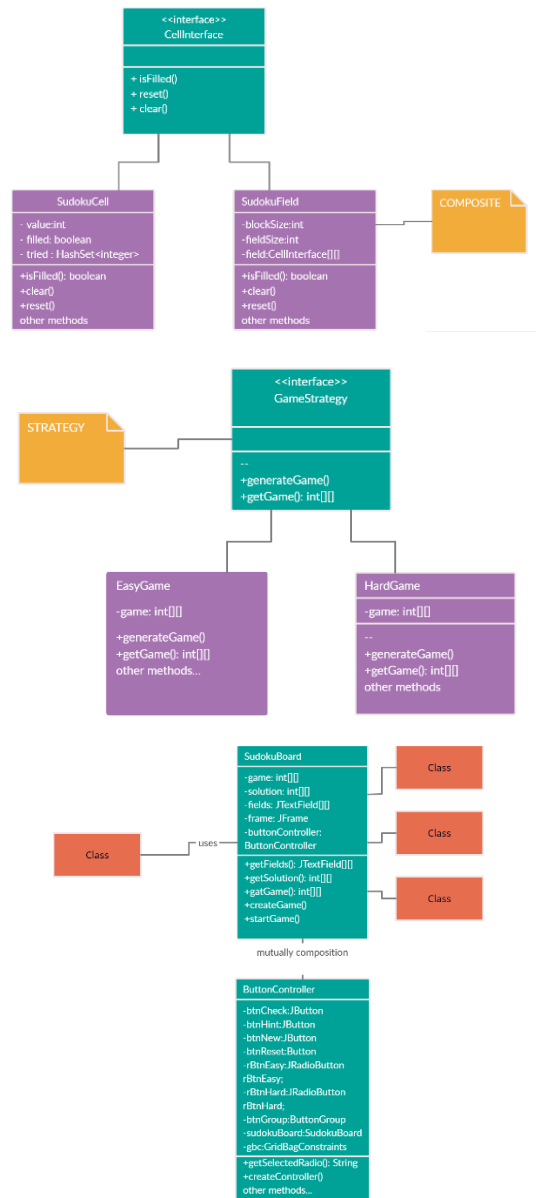
Sembilan kotak terdiri atas 3 bilangan acak angka 1-9 yang berbeda. Z_3 pada blok 9 tidak boleh sama dengan

bilangan acak pada setiap baris ketujuh dan setiap kolom kesembilan. Z_5 pada blok 9 tidak boleh sama dengan bilangan acak pada setiap baris kedelapan dan setiap kolom kedelapan. Z_9 tidak boleh sama dengan bilangan acak pada setiap baris kesembilan dan setiap kolom kesembilan.

Z_3	3
Z_5	8
Z_9	5

A. Flowchart dan Code

Berikut merupakan *flowchart* dari implementasi *linier congruential generator* pada Sudoku.



Gambar 1.1 Flowchart Gim Sudoku
Sumber: [Sudoku Game | Creately](#)

Implementasi level atau tingkatan permainan Sudoku klasik sebagai berikut.

1. Level mudah

	8	3		7				
9				4				
2		4				7		8
3			4	6		5	1	
				3	2		4	
8		6	7		5	3	9	2
	1	8		9	7			3
5	3		6		4		8	
4	2		3	8				5

2. Level medium

8		4					6	1
	3	7		4		8	2	
1					6			
2		5		3	4			
	8		9		2	4	5	3
		6	8	5		9	1	
9	1	2	4		5	6		
	6							5
5		8		1				

3. Level sulit

9	3		2			6	7	8
			4	7				2
5			6					
		8				2		
3	7		9		2	8	4	
	5	2		4			1	
			1					
	8	5	7		9			
		3	5	6	1	9		

Rancangan awal membuat kotak berukuran 9x9 yang terdiri atas 9 blok 3x3 dalam bahasa pemrograman.

```
#create grid
# 10 garis horizontal and 10 garis vertikal
for i in range(0,10):
    if i%3==0:
        #pemisahan blok 3x3
        pygame.draw.line(window, black, (50+50*i,50), (50+50*i,500) , 5)
        pygame.draw.line(window, black, (50,50+50*i), (500,50+50*i) , 5)
    else:
        #garis vertikal
        pygame.draw.line(window, black, (50+50*i,50), (50+50*i,500) , 3)
        #garis horizontal
        pygame.draw.line(window, black, (50,50+50*i), (500,50+50*i) , 3)
```

Berikut merupakan implementasi algoritma pembangkit bilangan acak berbasis kekongruenan linier atau LCG dalam bahasa pemrograman python.

```
#linier congruential generator (LCG)
for x in range(0,len(board[0])):
    for y in range(0,len(board[0])):
        #angka 1-9
        if(board[x][y]>0 and board[x][y]<10):
            val = Board_font.render(str(board[x][y]), True, (100,100,200) )
            #formula LCG
            window.blit(val,((y+1)*50 + 15, (x+1)*50 + 5))
```

V. KESIMPULAN

Matematika diskrit dapat diimplementasikan pada berbagai hal dalam berbagai bidang, termasuk bidang teknologi. Bilangan acak dapat dibangkitkan dengan program yang mengimplementasikan pembangkit bilangan acak semu Linear Congruential Generator (LCG) dalam pemilihan bilangan untuk persoalan gim Sudoku, sehingga dapat dibuat soal dengan teka-teki angka acak yang disusun sedemikian rupa untuk tujuan hiburan.

VI. SARAN

Dengan makalah aplikasi pembangkit bilangan acak berbasis kekongruenan linier ini, diharapkan materi matematika diskrit telah diimplementasikan dengan baik dan dapat dikembangkan. Pada saat ini, penulis membuat program permainan Sudoku sederhana untuk pengimplementasian pembangkitbilangan acak. Program pembangkit bilangan acak dapat dikembangkan lebih lanjut dengan turut serta mengubah gaya penulisan kata dan tata kalimat, atau menambah fitur menerjemahkan tulisan leet speak menjadi tulisan dengan kaidah penulisan yang baik dan benar. Selain itu, program ini juga dapat dimodifikasi lebih lanjut.

III. UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Tuhan Yang Maha Esa karena atas berkat rahmat dan karunia-Nya makalah yang berjudul “Aplikasi Pembangkit Bilangan Acak Berbasis Kekongruenan Lanjar Pada Gim Sudoku” dapat diselesaikan dengan baik. Penulis juga berterima kasih kepada Ibu Rinaldi Munir selaku dosen pengampu mata kuliah IF2120 Matematika Diskrit Semester I 2022/2023 Kelas K3

References

- [1] Munir, Rinaldi. 2015. Teori Bilangan. Bahan kuliah IF2120 Matematika Diskrit Semester I Tahun 2022/2023, Institut Teknologi Bandung.
- [2] Munir, Rinaldi. 2015. Pembangkit Bilangan Acak. Bahan kuliah IF3058 Kriptografi Semester I Tahun 2022/2023, Institut Teknologi Bandung.
- [3] Stuart, Andrew. 2007. Sudoku Creating and Grading. https://www.sudokuwiki.org/Sudoku_Creation_and_Grading.pdf
- [4] Melvin. 2021. “Create Sudoku Game Easily”. <https://sudokuessentials.com/create-sudoku>
- [5] Krisdiawan, Rio. 2018. Implementasi Model Pengembangan Sistem GDLC dan Algoritma Linear Congruential Generator Pada Game PUZZLE. <https://journal.uniku.ac.id/index.php/ilkom/article/view/1634/1211>

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 Desember 2022



Kartini Copa, 13521026