

Aplikasi Pembangkit Bilangan Acak Berbasis Kekongruenan Lanjar Pada Mesin *Slot*

Brianaldo Phandiarta - 13520113¹

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia

¹13520113@std.stei.itb.ac.id

Abstrak—Mesin *slot* merupakan salah satu permainan judi yang memberikan keuntungan terbesar bagi bandar. Keuntungan tersebut didapatkan dari manipulasi yang dilakukan oleh bandar. Salah satu cara termudah untuk melakukan manipulasi mesin *slot* adalah menggunakan 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 mesin *slot*.

Keywords—LCG, Mesin *Slot*, Teori Bilangan, Manipulasi.

I. PENDAHULUAN

Manusia mendapatkan kesenangan dari berbagai hiburan. Salah satu hiburan yang sudah ada sejak zaman dahulu adalah perjudian. Hal ini ditandai dengan ditemukannya berbagai peninggalan sejarah yang menandakan kemunculan judi lotre pada tahun 2300 sebelum masehi, permainan dadu pada tahun 500 sebelum masehi, dan permainan kartu pada tahun 800 masehi.

Perjudian adalah permainan dengan para pemainnya bertarung untuk memilih satu pilihan di antara banyak pilihan. Perjudian memberikan kebebasan kepada para pemainnya untuk memilih pilihan mereka, contohnya perjudian pada permainan dadu. Pada perjudian permainan dadu, bandar—*orang yang menyelenggarakan perjudian*—akan mengocok dadu. Dadu tersebut akan ditutup dengan tujuan agar dadu tidak dapat terlihat oleh para pemain. Kemudian, para pemain akan menebak angka ataupun jenis angka (ganjil atau genap) pada muka dadu yang menghadap ke atas. Pemain yang berhasil menebak memenangkan permainan tersebut.



Gambar 1. Ilustrasi Perjudian Dadu

Sumber: *shutterstock*

Selain itu, terdapat sistem perjudian lain, yaitu undian. Dalam sistem undian, berkebalikan dengan sistem perjudian pada umumnya, pilihan dari para pesertanya akan ditentukan secara acak. Pertama, para peserta akan diberikan tiket yang berisi nomor undian mereka. Kemudian, untuk menentukan pemenangnya, pihak penyelenggara akan menggunakan *random number generator* (RNG) atau pembangkit bilangan acak. Beberapa pihak penyelenggara juga menggunakan *pseudorandom number generator* (PRNG) atau *deterministic random bit generator* dengan tujuan memanipulasi pemenang undian yang akan dibahas lebih lanjut pada makalah ini.

Terdapat juga judi lotre yang merupakan gabungan konsep dari perjudian dan undian. Pada judi lotre, pembeli tiket lotre akan menentukan tujuh buah angka yang akan dibelinya. Kemudian, penyelenggara judi lotre akan mengumumkan tujuh buah angka lotre yang menentukan pemenang lotre. Ketujuh buah angka lotre tersebut dapat didapatkan dari RNG ataupun PRNG.

Seperti hiburan lainnya, seiring perkembangan teknologi, perjudian juga mengalami perkembangan. Contohnya, perkembangan perjudian dadu. Pada awal ditemukannya perjudian dadu, pengocokan dadu dilakukan secara langsung oleh bandar. Seiring perkembangan teknologi, perjudian dadu mulai menggunakan mesin untuk mengocok dadu. Sekarang, perjudian dadu bahkan dapat diselenggarakan secara daring melalui laman-laman perjudian daring.

Para penyelenggara judi tentunya tidak ingin mengalami kerugian. Oleh karena itu, banyak manipulasi hasil perjudian yang dilakukan oleh para penyelenggara. Hal ini semakin mudah untuk dilakukan seiring perkembangan teknologi perjudian.

Salah satu perjudian baru yang muncul ketika masa perkembangan teknologi adalah mesin *slot*. Mesin *slot* merupakan salah satu mesin perjudian yang menguntungkan bagi kasino. Identik dengan sistem perjudian lotre, mesin *slot* menggunakan gabungan sistem perjudian dan undian. Pada mesin *slot*, terdapat tiga *slot* yang harus dihentikan oleh pemain ketika mesin *slot* sedang bergerak. Ketika pemain mendapatkan tiga pola yang sama, maka pemain akan memenangkan. Ketika menghentikan mesin *slot*, pemain diberikan pilihan untuk menentukan waktu memberhentikan mesin. Sementara itu, mesin akan menggunakan sistem undian dalam menentukan pola yang akan dipilih.

Dalam pengimplementasiannya, kita dapat memanipulasi

keluaran dari pola mesin *slot*. Pada makalah ini, manipulasi keluaran dari pola mesin *slot* akan menggunakan pembangkit bilangan acak berbasis kekongruenan linier. Pembangkit bilangan acak berbasis kekongruenan linier menggunakan konsep teori bilangan yang merupakan salah satu konsep dalam matematika diskrit.

II. LANDASAN TEORI

A. Teori Bilangan

Teori Bilangan adalah cabang dari matematika murni yang mempelajari bilangan bulat dan sifatnya. Teori bilangan sangat bermanfaat dan banyak digunakan dalam *computer science* 'ilmu komputer'.

Bilangan bulat atau *integers* dapat direpresentasikan dalam basis lebih besar dari satu. Basis bilangan bulat yang umum digunakan adalah binari (basis 2), oktal (basis 8), desimal (basis 10), dan heksadesimal (basis 16). Secara formal, bilangan bulat dapat dinyatakan sebagai berikut,

Teorema 1 Misalkan b merupakan bilangan bulat lebih besar dari 1. Kemudian jika n merupakan bilangan bulat positif, n dapat dinyatakan dengan

$$n = a_k b^k + a_{k-1} b^{k-1} + \dots + a_1 b_0 + a_0,$$

dengan k adalah bilangan bulat nonnegatif, a_0, a_1, \dots, a_k adalah bilangan nonnegatif yang lebih kecil dari b , dan $a_k \neq 0$.

Dari Teorema 1, kita dapat menulis suatu bilangan bulat positif dengan notasi $(a_k a_{k-1} \dots a_1 a_0)_b$, dengan b merupakan basis dari $a_k a_{k-1} \dots a_1 a_0$.

Operasi utama yang menjadi landasan pada teori bilangan adalah pembagian habis dan modulus. Secara formal, pembagian habis dapat didefinisikan sebagai berikut,

Definisi 1 Jika a dan b adalah bilangan bulat dengan $a \neq 0$, a dapat membagi habis b jika terdapat bilangan bulat c yang memenuhi $b = ac$. Jika a dapat membagi habis b , a merupakan faktor atau pembagi dari b dan b merupakan pengali dari a . Notasi $a | b$ berarti a membagi habis b . Notasi $a \nmid b$ berarti a tidak membagi habis b .

Secara formal, sifat dari pembagian habis dapat dinyatakan sebagai berikut,

Teorema 2 Misalkan a , b , dan c adalah bilangan bulat, dengan $a \neq 0$. Maka,

- Jika $a | b$ dan $a | c$, $a | (b + c)$;
- Jika $a | b$, $a | bc$ untuk semua c ;
- Jika $a | b$ dan $b | c$, $a | c$.

Modulus merupakan operasi untuk mendapatkan sisa dari hasil pembagian. Secara formal, modulus dapat didefinisikan sebagai berikut,

Definisi 2 Misalkan a merupakan bilangan bulat dan d merupakan bilangan bulat positif. Maka, terdapat bilangan bulat q dan r yang unik dengan $0 \leq r < d$ sehingga $a = dq + r$. Dengan d disebut sebagai pembagi, a disebut sebagai terbagi, q disebut sebagai hasil bagi, dan r disebut sebagai sisa. Hasil bagi dan sisa dapat dinotasikan sebagai,

$$q = a \text{ div } d, r = a \text{ mod } d.$$

Pada bilangan bulat, dikenal dengan adanya bilangan prima. Secara informal, bilangan prima merupakan bilangan yang lebih besar dari 1 dan hanya dapat dibagi dengan 1 dan bilangan itu sendiri. Secara formal, bilangan prima dapat didefinisikan sebagai berikut,

Definisi 3 Suatu bilangan bulat p dikatakan prima jika dan hanya jika faktor dari p merupakan 1 dan p . Bilangan bulat positif yang lebih besar dari 1 dan bukan prima disebut sebagai bilangan komposit.

Menurut *The Fundamental Theorem of Arithmetic*,

Teorema 3 Semua bilangan bulat yang lebih besar dari 1 dapat ditulis secara unik dari sebuah bilangan prima atau perkalian dari dua atau lebih bilangan prima dengan faktor prima yang ditulis dengan urutan membesar.

B. Faktor Persekutuan Terbesar

Faktor persekutuan terbesar (FPB) adalah bilangan bulat terbesar yang dapat membagi dua buah bilangan bulat. Secara formal, FPB dapat didefinisikan sebagai berikut,

Definisi 4 Misalkan a dan b merupakan bilangan bulat bukan nol. Terdapat suatu bilangan bulat d dengan $d | a$ dan $d | b$ yang disebut sebagai faktor persekutuan terbesar dari a dan b . Faktor persekutuan terbesar dinotasikan sebagai FPB(a, b).

Jika dua buah bilangan bulat yang berbeda saling relatif prima, faktor persekutuan terbesar akan selalu bernilai 1. Sehingga, diperoleh definisi sebagai berikut,

Definisi 5 Suatu himpunan bilangan bulat a_1, a_2, \dots, a_n saling relative prima jika FPB(a_i, a_j) dengan $1 \leq i < j \leq n$.

Menurut Teorema Bézout, suatu faktor persekutuan terbesar dari dua bilangan dapat dibuat menjadi kombinasi linear dari dua bilangan tersebut. Secara formal, teorema tersebut dapat dinyatakan sebagai berikut,

Teorema 4 Jika a dan b merupakan bilangan bulat positif, terdapat bulangan bulat s dan t yang memenuhi

$$\text{FPB}(a, b) = sa + tb.$$

Dari Definisi 5 dan Teorema 4, dapat ditarik lema sebagai berikut,

Lema 1 Jika p adalah bilangan prima dan $p | a_1 a_2 \dots a_n$, dengan a_i merupakan bilangan bulat. $p | a_i$ untuk beberapa i .

Dari Lema 1, dapat ditarik kesimpulan bahwa jika sebuah bulangan bulat relatif prima terhadap modulusnya, dua bilangan bulat yang kongruen dapat dibagi sehingga menghasilkan kekongruenan yang valid. Sehingga diperoleh teorema sebagai berikut,

Teorema 5 Misalkan m adalah bilangan bulat positif dan a, b , dan c adalah bilangan bulat. Jika $a \equiv bc \pmod{m}$ dan $\text{FPB}(c, m) = 1$, $a \equiv b \pmod{m}$.

C. Sistem Kekongruenan Lanjar

Suatu sistem kekongruenan linier terdiri dari lebih dari satu

kekongruenan. Secara notasi, kekongruenan lanjut dapat ditulis sebagai $ax \equiv b \pmod{m}$. Dari Teorema 4, dapat diturunkan suatu teorema yang dapat membantu penyelesaian sistem kekongruenan lanjut sebagai berikut,

Teorema 6 Jika a dan m adalah bilangan bulat yang saling relative prima dengan $m > 1$, akan terdapat balikan dari $a \pmod{m}$ dan hasil dari modulo tersebut merupakan bilangan bulat unik.

D. Pembangkit Bilangan Acak

Pembangkit bilangan acak atau *random number generator* (RNG) adalah 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 *pseudorandom 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 *seed*. Terdapat berbagai macam pembangkit bilangan acak semu, salah satunya pembangkit bilangan acak berbasis kekongruenan lanjut yang memanfaatkan konsep teori bilangan.

E. Pembangkit Bilangan Acak Berbasis Kekongruenan Lanjar

Pembangkit bilangan acak berbasis kekongruenan lanjut atau *linear congruential generator* (LCG) adalah sebuah algoritma yang menghasilkan urutan dari bilangan acak semu yang dihitung dari persamaan diskontinu lanjut piecewise. Metode ini merupakan salah satu metode tertua dan terbaik yang dapat menghasilkan bilangan acak semu.

Teorema 7 Persamaan pembangkit bilangan acak semu berbasis kekongruenan lanjut adalah sebagai berikut,

$$x_{n+1} = (ax_n + c) \pmod{m} \quad (1)$$

dengan m merupakan modulus, a merupakan *'multiplier' faktor pengali*, c merupakan *'increment' faktor penambah*, dan x_0 merupakan *seed* atau bilangan awal. Persamaan (1) harus memenuhi syarat $2 \leq a < m$, $0 \leq c < m$, dan $0 \leq x_0 < m$.

E. Algoritma Pembangkit Bilangan Acak Berbasis Kekongruenan Lanjar

Berikut merupakan algoritma sederhana dari pembangkit bilangan acak berbasis kekongruenan lanjut yang ditulis dalam

bentuk notasi algoritma,

```

procedure LCG (input a, m, c, seed : integer,
                output rand : array of integer)
{ I.S. m, c, dan seed terdefinisi
  F.S. rand berisi bilangan-bilangan bulat yang
    dapat dihasilkan oleh LCG sesuai dengan
    urutannya }

KAMUS
{ Variabel }
curr : integer
{ Fungsi Antara }
function isIn (L : array of integer,
               x : integer) → boolean
{ Mengembalikan true jika x merupakan member
  dari L dan false jika bukan merupakan
  member dari L }
procedure append (input/output L : array of
                  integer,
                  x : integer)
{ I.S. L terdefinisi dan mungkin kosong;
  x terdefinisi
  F.S. x ditambahkan sebagai elemen
    terakhir pada L }

ALGORITMA
curr ← seed
while (not isIn(rand, curr)) do
  append(rand, curr)
  curr ← (a * curr + c) mod m
    
```

III. DESKRIPSI MASALAH

Mesin *slot* dikenal juga dengan variasi mesin buah, *puggy, the slots, mesin poker/pokies, fruities, dan slots*. Mesin *slot* adalah sebuah mesin perjudian yang menciptakan sebuah permainan peluang untuk pemainnya. Pada umumnya, mesin *slot* selalu dimanipulasi sedemikian sehingga sangat tidak mungkin memenangkannya. Mesin *slot* juga dikenal secara pejoratif sebagai 'bandit satu-tangan' karena lengan mekanikal besarnya yang ditempatkan pada sisi dari versi mesin mekanikal awalnya.



Gambar 2. Barisan Mesin Slot Bandar Udara Internasional McCarran Las Vegas
Sumber: wikipedia

Terdapat banyak jenis mesin *slot*. Jenis mesin *slot* yang paling populer adalah mesin *slot* bergaris jamak. Berbeda dengan mesin *slot* sederhana, mesin *slot* jamak memungkinkan pemainnya untuk memenangkan permainan tidak hanya ketika pemain mendapatkan pola garis horizontal, namun juga ketika pemainnya berhasil mendapatkan pola garis diagonal. Untuk penyederhanaan, mesin *slot* yang akan diimplementasikan pada makalah ini adalah mesin *slot* sederhana.

Mesin *slot* memiliki tiga *slot* yang akan diisi secara acak

ketika pemain menarik tuas atau menekan tombol. Tiga *slot* itu dapat berisi berbagai kombinasi simbol. Setiap mesin memiliki banyak simbol yang berbeda-beda tergantung desain masing-masing. Pada makalah ini, mesin *slot* akan memiliki delapan simbol yang berbeda yang direpresentasikan dalam bentuk bilangan bulat 0 sampai 7.

Dari setiap simbol yang terdapat pada mesin *slot*, terdapat satu simbol jackpot yang memberikan hadiah berkali-kali lipat kepada pemenangnya. Pada makalah ini, angka tujuh akan menjadi jackpot dari mesin *slot* yang akan diimplementasikan.

IV. ANALISIS DAN IMPLEMENTASI

A. Dekomposisi Masalah dan Alur Program

Sesuai dengan deskripsi masalah pada Bab III, dibutuhkan tiga buah pembangkit bilangan acak semu berbasis kekongruenan lanjar. Sebenarnya, dalam implementasinya, penulis dapat hanya menggunakan satu buah pembangkit bilangan acak semu berbasis kekongruenan lanjar dan dapat dipastikan pemain tidak akan dapat memenangkan permainan. Implementasi sederhana dengan menggunakan satu buah pembangkit bilangan acak semu berbasis kekongruenan lanjar adalah sebagai berikut,

```

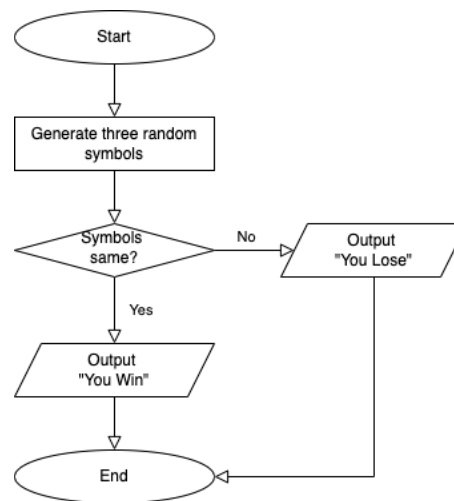
Program ImplementasiSederhana
{ Implementasi mesin slot sederhana menggunakan
  satu buah LCG }
KAMUS
{ Variabel }
time : integer
a, m, c, seed : integer
rand : array[0..2] of integer
{ Fungsi Antara }
procedure LCG3 (input a, m, c,
  seed : integer,
  output rand :
  array[0..2] of integer)
{ I.S. m, c, dan seed terdefinisi
  F.S. rand berisi tiga bilangan bulat
  pertama
  yang dihasilkan oleh LCG }
function getSecond () → integer
{ Mengembalikan waktu sekarang dalam dtk }
function isAllSame (L : array of integer)
  → boolean
{ Mengembalikan true jika semua elemen sama
  false jika ada elemen yang berbeda }
ALGORITMA UTAMA
{ Inisialisasi a, m, c }
a ← 2; m ← 8; c ← 0
time ← getSecond()
seed ← time mod m
{ Memanggil LCG3 }
LCG3 (a, m, c, seed, rand)
{ Jika diperhatikan, dengan parameter
  di atas, LCG tidak mungkin menghasilkan
  bilangan sama berturut-turut }
{ Check }
if (isAllSame(rand)) then
  output("You Win")
else
  output("You Lose")
  
```

```

{ Implementasi Fungsi Antar }
procedure LCG3 (input a, m, c, seed : integer,
  output rand : array[0..2] of integer)
{ I.S. m, c, dan seed terdefinisi
  F.S. rand berisi tiga bilangan bulat pertama
  yang dihasilkan oleh LCG }
KAMUS LOKAL
-
ALGORITMA
rand0 ← (a * seed + c) mod m
i traversal [1..2]
  randi ← (a * randi-1 + c) mod m
  
```

Akan tetapi, agar pengimplementasian dapat mendekati mesin *slot* sebenarnya, penulis akan membuat berbagai implementasi yang hanya memanipulasi agar pemain tidak dapat mendapatkan hadiah jackpot. Sehingga, pemain tetap dapat menang jika mendapatkan tiga simbol beruntun selain simbol 7.

Pertama, program akan menghasilkan tiga simbol (bilangan bulat dari 0 sampai 7) dari tiga buah pembangkit bilangan acak semu berbasis kekongruenan lanjar. Pembangkit bilangan acak semu berbasis kekongruenan lanjar tersebut akan dibangkitkan dengan menggunakan *seed* yang diperoleh dari waktu (dalam detik) ketika mesin dioperasikan oleh pemain. Tiga simbol itu akan disimpan pada suatu *array*. Kemudian program akan memeriksa semua elemen pada *array*. Jika seluruh elemen sama, maka pemain memenangkan permainan.



Gambar 3. Diagram Alir
Sumber: Dokumen pribadi

B. Implementasi I

Sesuai dengan Teorema 7, terdapat empat parameter yang harus ditentukan untuk setiap pembangkit bilangan acak semu berbasis kekongruenan lanjar, yaitu modulus, *multiplier*, *increment*, dan *seed*. Pada Implementasi I, penulis akan memanfaatkan sifat modulo untuk memanipulasi mesin *slot*.

Dalam menentukan parameter modulus, parameter lain selain modulus akan dibuat konstan sehingga dapat terlihat pola yang terbentuk. Sesuai dengan syarat pada Teorema 7, parameter konstan yang akan digunakan adalah $a = 2$, $c = 2$, dan $x_0 = 0$. Parameter modulus akan dicoba dalam *range* 3 sampai 10. Sehingga diperoleh,

Tabel 1. Parameter Modulus

m	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
3	2	0	2	0	2	0	2	0	2	0
4	2	2	2	2	2	2	2	2	2	2
5	2	1	4	0	2	1	4	0	2	1
6	2	0	2	0	2	0	2	0	2	0
7	2	6	0	2	6	0	2	6	0	2
8	2	6	6	6	6	6	6	6	6	6
9	2	6	5	3	8	0	2	6	5	3
10	2	6	4	0	2	6	4	0	2	6

Sumber: Dokumen Pribadi

Dari Tabel 1, dapat disimpulkan bahwa modulus berguna sebagai batas atas dari *range* bilangan acak yang dapat dihasilkan. Hal ini dapat terlihat jelas pada $m = 3, 5, 7$, dan 9 . Sehingga kita mendapatkan dua parameter modulus yang dapat digunakan, yaitu $m = 8$ sebagai parameter yang tidak dimanipulasi dan $m = 7$ sebagai parameter yang dimanipulasi.

Selain itu, dapat disimpulkan juga melalui (1) bahwa modulus merupakan operator yang terakhir digunakan. Sehingga, hasil dari pembangkit bilangan acak semu berbasis kekongruenan lanjar pasti memiliki hasil dengan *range* 0 hingga $m - 1$.

Sehingga, mesin *slot* dapat diimplementasikan dengan menggunakan dua buah pembangkit bilangan acak berbasis kekongruenan lanjar dengan $m = 8$ dan satu buah pembangkit bilangan acak berbasis kekongruenan lanjar dengan $m = 7$. Parameter selain modulo dan *seed* dapat dibebaskan. Parameter *seed* akan didapatkan dari waktu (dalam detik) ketika dioperasikan oleh pemain. Implementasi I dapat dilihat sebagai berikut,

```

Program Implementasi I
{ Implementasi I mesin slot }
KAMUS
  { Variabel }
  time : integer
  a, m, c : integer
  seed1, seed2, seed3 : integer
  rand : array[0..2] of integer
  { Fungsi Antara }
  function LCG1 (a, m, c, seed : integer)
    → integer
  { Mengembalikan nilai pertama yang
    dihasilkan }
  function getSecond () → integer
  { Mengembalikan waktu sekarang dalam dtk }
  function isAllSame (L : array of integer)
    → boolean
  { Mengembalikan true jika semua elemen sama
    false jika ada elemen yang berbeda }
ALGORITMA UTAMA
  { Inisialisasi a, c }
  a, c ← { bebas }
  time ← getSecond()
  seed1 ← (time + 1) mod m
  seed2 ← (time + 2) mod m
  seed3 ← (time + 3) mod m
  rand0 ← LCG1(a, 8, c, seed1)
  rand1 ← LCG1(a, 8, c, seed1)
  rand2 ← LCG1(a, 7, c, seed1)
  { Jika diperhatikan, dengan parameter
    di atas, LCG tidak mungkin menghasilkan
    tiga bilangan 7 }
    
```

```

  { Check }
  if (isAllSame(rand)) then
    output ("You Win")
  else
    output ("You Lose")
  { Implementasi Fungsi Antar }
  function LCG1 (a, m, c, seed : integer)
    → integer
  { Mengembalikan nilai pertama yang dihasilkan }
KAMUS LOKAL
  -
ALGORITMA
  → (a * seed + c) mod m
    
```

C. Implementasi II

Dengan menggunakan konsep gabungan dari Implementasi Sederhana dan Implementasi I, penulis akan membuat implementasi baru. Implementasi baru akan memanfaatkan parameter dari pembangkit bilangan acak semu berbasis kekongruenan lanjar yang dimanipulasi sehingga tidak akan menghasilkan bilangan 7 tanpa memanipulasi modulusnya.

Pada implementasi ini, dua buah pembangkit bilangan acak semu berbasis kekongruenan lanjar akan mendapatkan *seed* dari waktu ketika mesin *slot* dioperasikan dan satu buah pembangkit bilangan acak semu berbasis kekongruenan lanjar akan mendapatkan *seed* yang telah ditentukan. Hal ini dilakukan karena tidak mungkin didapatkan sebuah persamaan (1) dengan *seed* acak yang tidak dapat menghasilkan angka 7. Tujuan penulis adalah menemukan parameter *a* atau *multiplier*, x_0 atau *seed*, dan *c* atau *increment* sedemikian sehingga pembangkit bilangan acak semu berbasis kekongruenan lanjar menghasilkan suatu himpunan dengan elemen unik sebanyak mungkin tanpa elemen bernilai 7.

Pertama, penulis akan menentukan kombinasi dari seluruh parameter pembangkit bilangan acak semu berbasis kekongruenan lanjar. Dari Teorema 7, terdapat 6 kemungkinan *multiplier* ($2 \leq a < 8$), terdapat 8 kemungkinan *increment* ($0 \leq c < 8$), dan 8 kemungkinan *increment* ($0 \leq c < 8$). Sehingga terdapat 384 kombinasi.

Penulis akan menggunakan *seed* sebagai parameter konstan agar mempermudah pencarian parameter yang sesuai. Sehingga, diperoleh hasil sebagai berikut.

Tabel 2. Parameter *Konstan* $x_0 = 0$

x_0	a	c	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
0	2	0	0	0	0	0	0	0	0	0
		1	1	3	7	7	7	7	7	7
		2	2	6	6	6	6	6	6	6
		3	3	1	5	5	5	5	5	5
		4	4	4	4	4	4	4	4	4
		5	5	7	3	3	3	3	3	3
		6	6	2	2	2	2	2	2	2
	7	7	5	1	1	1	1	1	1	
3	0	0	0	0	0	0	0	0	0	

	4	1	1	4	5	0	1	4	5	0
		2	2	0	2	0	2	0	2	0
		3	3	4	7	0	3	4	7	0
		4	4	0	4	0	4	0	4	0
		5	5	4	1	0	5	4	1	0
		6	6	0	6	0	6	0	6	0
		7	7	4	3	0	7	4	3	0
	5	0	0	0	0	0	0	0	0	0
		1	1	5	5	5	5	5	5	5
		2	2	2	2	2	2	2	2	2
		3	3	7	7	7	7	7	7	7
		4	4	4	4	4	4	4	4	4
		5	5	1	1	1	1	1	1	1
		6	6	6	6	6	6	6	6	6
	6	0	0	0	0	0	0	0	0	0
		1	1	6	1	5	5	5	5	5
		2	2	4	2	2	2	2	2	2
		3	3	2	3	7	7	7	7	7
		4	4	0	4	4	4	4	4	4
		5	5	6	5	1	1	1	1	1
		6	6	4	6	6	6	6	6	6
	7	0	0	0	0	0	0	0	0	0
		1	1	7	3	3	3	3	3	3
		2	2	6	6	6	6	6	6	6
		3	3	5	1	1	1	1	1	1
		4	4	4	4	4	4	4	4	4
		5	5	3	7	7	7	7	7	7
		6	6	2	2	2	2	2	2	2
8	0	0	0	0	0	0	0	0	0	
	1	1	0	1	0	1	0	1	0	
	2	2	0	2	0	2	0	2	0	
	3	3	0	3	0	3	0	3	0	
	4	4	0	4	0	4	0	4	0	
	5	5	0	5	0	5	0	5	0	
	6	6	0	6	0	6	0	6	0	
7	7	0	7	0	7	0	7	0		

Sumber: Dokumen Pribadi

Tabel 3. Parameter Konstan $x_0 = 5$

x_0	a	c	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
5	2	0	2	4	0	0	0	0	0	0
		1	3	7	7	7	7	7	7	7
		2	4	2	6	6	6	6	6	6
		3	5	5	5	5	5	5	5	5
		4	6	0	4	4	4	4	4	4
		5	7	3	3	3	3	3	3	3
		6	0	6	2	2	2	2	2	2
	3	7	1	1	1	1	1	1	1	1
		0	7	5	7	5	7	5	7	5
		1	0	1	4	5	0	1	4	5
		2	1	5	1	5	1	5	1	5
		3	2	1	6	5	2	1	6	5
		4	3	5	3	5	3	5	3	5
		5	4	1	0	5	4	1	0	5
	4	6	5	5	5	5	5	5	5	5
		7	6	1	2	5	6	1	2	5
		0	4	0	0	0	0	0	0	0
		1	5	5	5	5	5	5	5	5
		2	6	2	2	2	2	2	2	2
		3	7	7	7	7	7	7	7	7
		4	0	4	4	4	4	4	4	4
	5	5	1	1	1	1	1	1	1	1
		6	2	6	6	6	6	6	6	6
		7	3	3	3	3	3	3	3	3
		0	1	5	4	0	0	0	0	0
		1	2	3	5	5	5	5	5	5
		2	3	1	6	2	2	2	2	2
		3	4	7	7	7	7	7	7	7
6	4	5	5	0	4	4	4	4	4	
	5	6	3	1	1	1	1	1	1	
	6	7	1	2	6	6	6	6	6	
	7	0	7	3	3	3	3	3	3	
	0	6	4	0	0	0	0	0	0	
	1	7	3	3	3	3	3	3	3	
	2	0	2	6	6	6	6	6	6	
7	3	1	1	1	1	1	1	1	1	
	4	2	0	4	4	4	4	4	4	
	5	3	7	7	7	7	7	7	7	

7	6	4	6	2	2	2	2	2	2
	7	5	5	5	5	5	5	5	5
	0	3	5	3	5	3	5	3	5
	1	4	5	4	5	4	5	4	5
	2	5	5	5	5	5	5	5	5
	3	6	5	6	5	6	5	6	5
	4	7	5	7	5	7	5	7	5
	5	0	5	0	5	0	5	0	5
	6	1	5	1	5	1	5	1	5
	7	2	5	2	5	2	5	2	5

Sumber: Dokumen Pribadi

Tabel 2 dan Tabel 3 merupakan representasi dari percobaan-percobaan yang dilakukan penulis. Pada Tabel 3, terdapat parameter yang menghasilkan jumlah elemen unik sebanyak 5 (5, 3, 1, 6, dan 2) yang dapat ditemukan oleh penulis. Parameter tersebut adalah $x_0 = 5$, $a = 5$, dan $c = 2$.

Sehingga, mesin slot dapat diimplementasikan sebagai berikut,

```

Program ImplementasiIII
{ Implementasi II mesin slot }
KAMUS
{ Variabel }
time : integer
a, m, c, seed : integer
rand : array[0..2] of integer
{ Fungsi Antara }
function getSecond () → integer
{ Mengembalikan waktu sekarang dalam dtk }
function isAllSame (L : array of integer)
→ boolean
{ Mengembalikan true jika semua elemen sama
false jika ada elemen yang berbeda }
function getRandom (L : array of integer)
→ integer
{ Mengembalikan salah satu elemen pada L
secara acak }
ALGORITMA UTAMA
{ Inialisasi a, c }
a, c ← { bebas }
time ← getSecond()
seed ← time mod m
rand0 ← getRandom(LCG(a, 8, c, seed))
rand1 ← getRandom(LCG(a, 8, c, seed))
rand2 ← getRandom(LCG(5, 8, 2, 5))
{ Jika diperhatikan, dengan parameter
di atas, LCG tidak mungkin menghasilkan
tiga bilangan 7 }
{ Check }
if (isAllSame(rand)) then
output("You Win")
else
output("You Lose")

```

V. KESIMPULAN DAN SARAN

Dari Analisis dan Implementasi, dapat disimpulkan bahwa pembangkit bilangan acak berbasis kekongruenan lanjar dapat membangkitkan bilangan acak. Selain itu, pembangkit bilangan acak berbasis kekongruenan lanjar juga dapat dimanipulasi

sehingga tidak dapat menghasilkan bilangan tertentu. Untuk penelitian lebih lanjut, implementasi mesin slot pembangkit bilangan acak semu berbasis kekongruenan lanjar dapat dibandingkan dengan menggunakan pembangkit bilangan acak semu lain. Selain itu, mesin slot juga dapat diimplementasikan berdasarkan probabilitas menggunakan pembangkit bilangan acak semu berbasis kekongruenan lanjar.

VI. UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Tuhan Yang Maha Esa karena atas berkat dan rahmat-Nya, penulis dapat menyelesaikan makalah ini. Penulis juga ingin mengucapkan terima kasih kepada Ibu Dr. Nur Ulfa Maulidevi, S.T, M.Sc. selaku dosen mata kuliah IF2120 Matematika Diskrit Kelas K3 serta Bapak Dr. Ir. Rinaldi Munir, M.T. yang menyediakan bahan ajar dan template makalah sehingga memudahkan penulis dalam Menyusun makalah ini. Penulis juga mengucapkan terima kasih kepada kedua orang tua dan keluarga penulis yang senantiasa mendoakan dan mendukung pembuatan makalah ini.

REFERENSI

- [1] Rosen, Kenneth H., "Discrete Mathematics and Its Applications", 7th ed., M. Lange, Ed. New York: McGraw-Hill, 2012, pp. 237-305.
- [2] Munir, Rinaldi, "Matematika Diskrit", 3rd ed., Bandung: Informatika Bandung, 2010, pp. 175-218.
- [3] Knutsson, Anders, "Usage of LCG/CLCG Numbers for Electronic Gambling Applications", Stockholm: Simovits Consulting, 2018.
- [4] Chambers, Kerry, "Gambling for Profit: Lotteries, Gaming Machines, and Casinos in Cross-national Focus", Toronto: U of Toronto Press, 2011.
- [5] Liem, Inggriani, "Diktat Pemrograman Prosedural", Bandung: IF-ITB, 2007.
- [6] Zakiah, Nena. "7 Fakta Sejarah Judi, Dikenal dari Ribuan Tahun Lalu", <https://www.idntimes.com/science/discovery/na-na-zakiah-1/fakta-sejarah-judi/2> diakses pada 13 Desember 2021.

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, 14 Desember 2021



Brianaldo Phandiarta - 13520113