

Analisis Penggunaan Algoritma Linear Congruential Generator Untuk Mendapatkan Rare Pack di Pokemon Trading Card Game Pocket

Muhammad Jordan Ferimeison - 13524047

Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail: ferimeison@gmail.com , 13524047@std.stei.itb.ac.id

Abstraksi—Pokemon TCG Pocket merupakan salah satu video game yang menggunakan sistem pembangkit bilangan acak semu (PRNG). Penelitian ini akan membahas penggunaan sistem pembangkit bilangan acak berbasis kekongruenan linear (LCG) pada Pokemon TCG Pocket. Apakah sistem tersebut dapat diimplementasikan untuk menentukan kemunculan Rare Pack pada Pokemon TCG Pocket.

Kata kunci—acak, Rare Pack, TCG, PRNG, LCG

I. PENDAHULUAN

Video game merupakan salah satu kegiatan yang dilakukan untuk mengisi waktu luang. Saat bermain video game, banyak aspek seru dan menarik yang membuat pemain terus bermain video game tersebut. Terkadang, aspek ‘acak’ digunakan sebagai aspek yang menarik pemain. Salah satu penggunaan aspek ‘acak’ adalah pada sistem membuka *Card Pack* dalam game Pokemon TCG Pocket.

Pokemon Trading Card Game Pocket adalah adaptasi digital dari game Pokemon Trading Card Game. Permainan ini berfokus pada mengumpulkan kartu, bertarung dengan kartu yang dimiliki, dan bertukar kartu dengan teman. Pokemon Trading Card Game Pocket telah diunduh lebih dari 60 Juta pemain di platform iOS dan Android [1].

Untuk mendapatkan kartu di Pokemon TCG Pocket, pemain perlu membuka Card Pack. Masing-masing Card Pack berisi 5 kartu secara acak. Pemain juga memiliki kesempatan 0.05% untuk mendapatkan Rare Pack, sebuah Card Pack yang berisi kartu di atas wajah 4. Sistem pembangkit bilangan acak semu (PRNG) yang digunakan oleh Pokemon TCG Pocket masih belum diketahui. Oleh karena itu, peneliti tertarik untuk menguji penggunaan algoritma pembangkit bilangan acak berbasis kekongruenan linear pada Pokemon TCG Pocket.

Algoritma pembangkit bilangan acak berbasis kekongruenan linear (LCG) adalah salah satu algoritma sederhana yang memanfaatkan sifat bilangan untuk menciptakan bilangan baru/ Sistem ini bekerja dengan memanfaatkan nilai awal, faktor pengali, faktor penambah, dan modulus.

Makalah ini akan membahas penggunaan algoritma pembangkit bilangan acak berbasis kekongruenan linear (LCG) untuk menentukan kemunculan Rare Pack pada Pokemon TCG Pocket.

II. DASAR TEORI

A. LCG

Sebelum memahami LCG, kita perlu memahami operasi modulo. Modulo biasa dinotasikan dalam bentuk *mod*. Operasi modulo adalah suatu operasi yang memenuhi persamaan

$$a \bmod m = r$$

$$a = m * q + r$$

Dengan q adalah suatu bilangan integer sembarang dan r berada di antara $[0..m-1]$ [2]. Misal,

$$5 \bmod 3 = 2$$

Karena memenuhi persamaan

$$5 = 3 * 1 + 2$$

Bilangan 2 berada di antara $[0..3-1]$, sehingga 2 adalah jawaban valid dari 5 modulo 3. Dengan kata lain, hasil dari modulo suatu bilangan adalah sisa dari pembagian bilangan tersebut.

$$a \bmod m = r \equiv a = m * q + r$$

LCG (Linear Congruential Generator) adalah pembangkit bilangan acak yang berbasis kekongruenan linear. Sistem persamaan ini memenuhi persamaan umum sebagai berikut [3],

$$X_n = (a * X_{n-1} + b) \bmod m$$

$$X_n = \text{bilangan suku ke } - n$$

$$X_{n-1} = \text{bilangan suku ke } - (n - 1)$$

$$a = \text{faktor pengali/multiplier}$$

$$b = \text{faktor penjumlah/increment}$$

$$m = \text{faktor pembagi/modulus}$$

Sistem persamaan ini masuk ke dalam kategori Pseudo-Random Number Generator (PRNG) atau Pembangkit Bilangan Acak Semu. Hal ini dikarenakan bilangan yang dibuat tidak sepenuhnya acak, berdasarkan pada suatu persamaan, dapat diprediksi, dan dapat berulang. Untuk menghasilkan suatu angka, sistem persamaan LCG membutuhkan suatu kunci pembangkit (*seed*) sebagai dasar dari deret yang akan dibentuk (X_0).

B. Kombinatorika

Kombinatorika adalah cabang matematika yang diperuntukkan untuk menghitung jumlah penyusunan objek tanpa mengenumerasi semua kemungkinan penyusunannya [4]. Dalam kombinatorika, terdapat kaidah dasar menghitung yaitu,

a. Kaidah Perkalian

Kaidah ini menjelaskan jika terdapat dua percobaan yang saling mengikat, maka peluang dari percobaan tersebut jika dilakukan secara bersamaan adalah hasil kali dari kedua percobaan tersebut

Percobaan 1 : p kemungkinan

Percobaan 2 : q kemungkinan

Percobaan 1 dan percobaan 2 : $p * q$ kemungkinan.

b. Kaidah Penjumlahan

Kaidah ini menjelaskan jika terdapat dua percobaan yang tidak saling mengikat, tetapi masih memiliki hubungan, maka peluang dari percobaan tersebut jika dilihat dari hubungannya adalah hasil penjumlahan dari kedua percobaan tersebut.

Percobaan 1 : p kemungkinan

Percobaan 2 : q kemungkinan

Percobaan 1 atau percobaan 2 : $p + q$ kemungkinan.

Selain kaidah dasar, dalam kombinatorika juga terdapat aturan permutasi. Permutasi adalah jumlah urutan berbeda dari pengurutan objek. Untuk itu, berlaku persamaan permutasi n buah objek sebagai

$$n! = n(n - 1)(n - 2) \dots 1$$

$P(n, r)$ berarti jumlah kemungkinan urutan berbeda r buah elemen yang dipilih dari n buah elemen. Permutasi memenuhi persamaan berikut

$$\frac{n!}{(n - r)!}$$

C. Pokemon Trading Card Games (TCG) Pocket

Pokemon Trading Card Game Pocket adalah adaptasi digital dari game Pokemon Trading Card Game. Permainan ini berfokus pada mengumpulkan kartu, bertarung dengan kartu yang dimiliki, dan bertukar kartu dengan teman. Untuk mendapatkan kartu baru, pemain perlu membuka Booster/Card Pack. Pada permainan ini, terdapat dua jenis Card Pack yang dapat pemain buka, yaitu Regular Pack dan Rare Pack. Kedua jenis Card Pack ini didapat secara acak seperti yang tertera berikut

Booster Pack Selection Probability	
Regular pack	99.950%
Rare pack	0.050%

Fig. 1. Booster/Card Pack Probability; Sumber: Arsip Penulis

Peluang untuk mendapatkan rare pack adalah 0.050%. Hal yang membedakan Rare Pack dengan Regular Pack adalah peluang mendapatkan kartu dengan rarity lebih tinggi.

Pokemon TCG Pocket membagi Rarity kartu menjadi (terurut dari yang paling mudah didapat)

a. Diamond (1 s.d. 4 Diamond)



Fig. 2. Diamond Rariry; Sumber: Arsip Penulis

b. Star (1 s.d. 3 Star)



Fig. 3. Star Rariry; Sumber: Arsip Penulis

c. Shiny (1 s.d. 2 Shining Star)



Fig. 4. Shiny Rariry; Sumber: Arsip Penulis

d. Crown (1 Crown)



Fig. 5. Crown Rariry; Sumber: Arsip Penulis

Rarity yang lebih tinggi dalam Pokemon TCG Pocket akan memberikan efek pertarungan yang lebih bagus. Kartu dengan rarity yang lebih tinggi juga dapat berarti kartu yang lebih kuat. Misalnya, kartu 4 Diamond adalah kartu EX, versi lebih kuat dari suatu varian pokemon. Sedangkan kartu 2 Star adalah versi "full art" dari kartu tersebut. Mendapat kartu dengan rarity tinggi berarti mendapatkan kartu kuat yang lebih bagus.

Pada Regular Pack, pemain akan mendapatkan 3 kartu Diamond (1 s.d. 3 diamond) dan 2 kartu acak dari rarity lainnya. Pada Rare Pack, pemain berkesempatan untuk mendapatkan 5 kartu dengan rarity star atau lebih tinggi. Hal ini yang membuat player menyebut Rare Pack dengan istilah "God Pack", karena rarity yang ada dalam Rare Pack berada minimal di 1 Star.

III. METODOLOGI

A. Menentukan persamaan

Persamaan LCG yang akan dibuat akan menghasilkan angka dalam range [0..99999]. Range ini dipilih karena total dari persentase yang terbentuk adalah 100.000%. Maka, terdapat 100000 angka dimulai dari angka 0. Hal ini direpresentasikan oleh table berikut

TABLE I. KEMUNGKINAN CARD PACK

No	Table Column Head		
	Kategori	Range	Persentase
1.	Regular	0 s.d. 99949	99.950%
2.	Rare	99950 s.d. 99999	0.050%

Tabel tersebut merupakan table distribusi pembagian pack yang ada dalam game Pokemon TCG Pocket. Pada percobaan ini, diasumsikan nilai yang dihasilkan oleh suatu algoritma PRNGs akan menentukan pack mana yang akan dibuka oleh player. Jika bilangan tersebut jatuh pada range Regular Pack, maka player akan mendapat Regular pack. Sebaliknya, jika player mendapat angka dalam range Rare Pack, player akan mendapat Rare Pack.

Selanjutnya, akan dibuat persamaan LCG yang memenuhi persamaan umum berikut

$$X_n = (a * X_{n-1} + b) \text{ mod } m \quad (1)$$

Untuk menyelesaikan persamaan, dibutuhkan beberapa nilai yaitu nilai a sebagai bilangan pengali, nilai b sebagai bilangan penjumlah, dan nilai m sebagai bilangan pembagi/modulo. Selain itu, dibutuhkan nilai X_0 atau *seed* sebagai dasar dari LCG.

Untuk menentukan nilai a , b dan X_0 akan digunakan 5 angka acak pada ID pemain pertama, digunakan asumsi 5 angka pertama dari ID sebagai nilai a sehingga diperoleh nilai a adalah 80916. Untuk nilai b , digunakan asumsi 5 angka ID berikutnya sehingga diperoleh nilai b adalah 89859. Dengan menggunakan asumsi nilai X_0 adalah 5 angka terakhir dari ID player, didapat nilai X_0 memenuhi

$$a = 80916$$

$$b = 89859$$

$$X_0 = 52143$$

Untuk nilai m , didapat dari jumlah semua bilangan yang mungkin muncul oleh algoritma PRNGs, yaitu jumlah range Regular pack + range rare pack,

$$m = 99950 + 50 = 100000$$

Sehingga, diperoleh nilai m adalah 100000. Setelah mendapat semua angka yang dibutuhkan, diperoleh persamaan LCG sebagai berikut,

$$X_n = (80916 * X_{n-1} + 89859) \text{ mod } 100000 \quad (2)$$

Untuk menentukan nilai dari n sehingga X_n memenuhi range Rare Pack, digunakan kode sebagai berikut untuk melakukan simulasi deret LCG.

```

iter ← 0
res ← X0
while (res < 99950 and iter < 100000) do
    res = ((a * res) + b) mod 100000
output(res, iter)

```

Fig. 6. Kode menentukan kemunculan Rare Pack pertama; Sumber: Arsip Penulis

Dari kode tersebut, diperoleh dua nilai sebagai berikut

$$X_n = 99951$$

$$n = 1587$$

Kedua nilai tersebut didapat setelah melakukan iterasi LCG dengan persamaan di atas dengan X_n adalah res dan n adalah iter.

Ini hanyalah salah satu dari kemungkinan deret LCG yang ada. Mari kita 'acak' angka yang ada untuk melakukan eksplorasi lebih dalam. Simulasi akan dilakukan dengan melakukan permutasi pada urutan nilai a , b dan X_0 dari angka yang didapat dari ID pemain.

Kemungkinan 2

$$a = 89859$$

$$b = 80916$$

$$X_0 = 52143$$

Dari kemungkinan 2, didapat nilai a , b dan X_0 yang merupakan bagian dari permutasi nilai a , b dan X_0 pada kemungkinan 1. Pada kemungkinan 2, diperoleh persamaan sebagai berikut

$$X_n = (a * X_{n-1} + b) \text{ mod } m \quad (3)$$

Dengan memasukkan persamaan tersebut ke dalam kode pada Fig. 6, diperoleh dua nilai sebagai berikut,

$$X_n = 99953$$

$$n = 437$$

Kemungkinan 3

$$a = 52143$$

$$b = 89859$$

$$X_0 = 80916$$

Dari kemungkinan 2, didapat nilai a , b dan X_0 yang merupakan bagian dari permutasi nilai a , b dan X_0 pada kemungkinan 1. Pada kemungkinan 2, diperoleh persamaan sebagai berikut

$$X_n = (a * X_{n-1} + b) \text{ mod } m \quad (4)$$

Dengan memasukkan persamaan tersebut ke dalam kode pada Fig. 6, diperoleh dua nilai sebagai berikut,

$$X_n = 99999$$

$$n = 203$$

Kemungkinan 4

$$a = 80916$$

$$b = 52143$$

$$X_0 = 89859$$

Dari kemungkinan 2, didapat nilai a , b dan X_0 yang merupakan bagian dari permutasi nilai a , b dan X_0 pada kemungkinan 1. Pada kemungkinan 2, diperoleh persamaan sebagai berikut

$$X_n = (a * X_{n-1} + b) \text{ mod } m \quad (5)$$

Dengan memasukkan persamaan tersebut ke dalam kode pada Fig. 6, diperoleh dua nilai sebagai berikut,

$$X_n = 99979$$

$$n = 1215$$

Kemungkinan 5

$$a = 52143$$

$$b = 80916$$

$$X_0 = 89859$$

Dari kemungkinan 2, didapat nilai a , b dan X_0 yang merupakan bagian dari permutasi nilai a , b dan X_0 pada kemungkinan 1. Pada kemungkinan 2, diperoleh persamaan sebagai berikut

$$X_n = (a * X_{n-1} + b) \text{ mod } m \quad (6)$$

Dengan memasukkan persamaan tersebut ke dalam kode pada Fig. 6, diperoleh dua nilai sebagai berikut,

$$X_n = 89859$$

$$n = 100000$$

Dari hasil tersebut, dapat ditarik kesimpulan bahwa pada persamaan ini, tidak ditemukan angka yang berada dalam range Rare Pack dalam 100000 iterasi. Artinya, persamaan ini tidak akan menghasilkan angka dalam range Rare Pack. Jika mendapat persamaan ini, maka pemain tidak akan mendapatkan Rare Pack setelah membuka 100000 Card Pack.

Kemungkinan 6

$$a = 89859$$

$$b = 52143$$

$$X_0 = 80916$$

Dari kemungkinan 2, didapat nilai a , b dan X_0 yang merupakan bagian dari permutasi nilai a , b dan X_0 pada

kemungkinan 1. Pada kemungkinan 2, diperoleh persamaan sebagai berikut

$$X_n = (a * X_{n-1} + b) \text{ mod } m \quad (7)$$

Dengan memasukkan persamaan tersebut ke dalam kode pada Fig. 6, diperoleh dua nilai sebagai berikut,

$$X_n = 99983$$

$$n = 1230$$

Ke-6 kemungkinan tadi hanyalah sebagian dari semua kemungkinan deret LCG yang mungkin terbentuk. Misalnya, jika berasumsi pemain telah membuka 1 Card Pack dan mendapatkan Regular Pack. Untuk itu, persamaan LCG yang mungkin terbentuk ada sebanyak

$$99950 \times 99950 \times \dots \times 99950 = 99950^1$$

Jika pemain telah membuka 2 Regular Pack, maka banyaknya persamaan LCG yang mungkin terbentuk ada sebanyak

$$99950 \times 99950 = 99950^2$$

Sekarang, bagaimana jika pemain membuka n buah Card Pack? Berapa kemungkinan LCG yang terbentuk jika kita tidak tahu ia mendapat Regular Pack atau Rare Pack? Sekarang kemungkinan kartu tiap pembukaan menjadi 100000, dari awalnya 99950 saat membuka Regular Pack. Sehingga, banyaknya persamaan LCG yang mungkin terbentuk adalah sebanyak

$$100000 \times 100000 \times \dots \times 100000 = 100000^n \quad (8)$$

Dengan n adalah jumlah Card Pack yang telah dibuka. Hal ini menandakan kemungkinan persamaan LCG yang dapat terbentuk akan semakin besar sejalan dengan jumlah pack yang dibuka.

IV. HASIL DAN PEMBAHASAN

Setelah melakukan percobaan tersebut, didapat kesimpulan bahwa jika kita mampu menentukan deret LCG yang tepat, kita dapat memprediksi kapan kemunculan Rare Pack pada deret tersebut. Misal, pada kemungkinan pertama kita dapat menghitung kemungkinan mendapat Rare Pack pada 100000 pembukaan Card Pack pertama dengan kode berikut

```

iter ← 0
res ← X0
while (iter < 100000) do
    res = ((a * res) + b) mod 100000
    if (res > 99949) then
        output(res, iter)
    
```

Fig. 7. Kode menentukan kemunculan Rare Pack; Sumber: Arsip Penulis

Setelah memasukkan persamaan ke kode tersebut, didapat hasil sebagai berikut,

TABLE II. KEMUNCULAN RARE PACK PERSAMAAN 2

No	Keterangan		
	Suku (n)	Nilai (Xn)	Selisih suku dengan setelahnya (n)
1.	1587	99951	323
2.	1910	99983	2802
3.	4712	99951	323
4.	5035	99983	2802
5.	7837	99951	323
6.	8160	99983	2802
7.	10962	99951	323
8.

Dari table 2, dapat disimpulkan bahwa deret akan berulang setiap 3125 suku.

$$2802 + 323 = 3125 \quad (2)$$

Tetapi, apakah deret hanya akan berulang pada kemungkinan 1, bagaimana dengan kemungkinan lainnya? Dengan memasukkan kemungkinan 2 ke kode Fig 7, diperoleh hasil sebagai berikut

TABLE III. KEMUNCULAN RARE PACK PERSAMAAN 3

No	Keterangan		
	Suku (n)	Nilai (Xn)	Selisih suku dengan setelahnya (n)
1.	437	99953	1000 $n = 437, \text{res} = 99953$
2.	1437	99953	1000
3.	2437	99953	1000
4.

Setelah melihat hasil yang dibentuk, kembali terjadi pola pada suku Rare Pack dihasilkan. Pada kemungkinan dua, Rare Pack akan didapat setiap 1000 pembukaan Card Pack.

Dari dua kemungkinan tersebut, didapati bahwa deret yang diciptakan oleh suatu persamaan LCG akan berulang setiap n kali. Perulangan ini berbeda-beda tergantung dengan nilai a , b dan m pada deret. Hal ini juga berarti angka yang dihasilkan oleh persamaan LCG tidak sepenuhnya acak, tapi angka semi acak (pseudo-random).

Selain itu, deret LCG juga dapat tidak menghasilkan angka pada rentang nilai tertentu. Pada persamaan 6, deret tidak menghasilkan nilai pada rentang Rare Pack.

V. KESIMPULAN DAN SARAN

Penggunaan algoritma LCG dapat dilakukan untuk mensimulasikan algoritma PRNGs yang ada dalam game. Algoritma ini merupakan salah satu algoritma yang dapat kita gunakan dalam penentuan angka acak untuk suatu hal. Algoritma ini dapat digunakan apabila kita berhasil menentukan angka yang diperlukan. Salah satu penggunaannya adalah dalam menghasilkan angka acak untuk sistem *gacha*. Seperti yang dilakukan dalam simulasi, algoritma ini dapat digunakan untuk memprediksi kapan kemunculan Rare Pack dalam game. Penggunaan LCG dapat digunakan untuk menentukan kapan deret akan menghasilkan angka pada range tertentu.

Namun, penggunaan deret LCG masih memiliki beberapa kelemahan seperti menghasilkan urutan angka yang berulang setelah jumlah iterasi yang tertentu, hingga angka tidak ada dalam hasil deret algoritma LCG. Kemungkinan ini perlu diminimalisir untuk memastikan aspek 'adil' dan 'seru' dalam suatu permainan. Optimalisasi seperti merubah bilangan pembangkit (*seed*) secara berkala atau penggunaan algoritma lain seperti MT, XOR, atau algoritma PRNGs lain dapat meningkatkan unsur 'acak' yang ingin dihasilkan. Perubahan ini juga dapat digunakan untuk memastikan angka yang diinginkan masuk ke dalam jangkauan hasil dari algoritma PRNG.

Oleh karena itu, penulis menyarankan melakukan perbandingan dengan menggunakan algoritma lain dalam menganalisis kemunculan Rare Pack pada Game Pokemon TCG Pocket untuk meningkatkan dampak dari penggunaan suatu algoritma PRNG. Melakukan simulasi secara nyata untuk menentukan apakah pada pembukaan ke- n akan mendapat Rare Pack juga dapat dilakukan untuk membuktikan kebenaran dari algoritma PRNG yang digunakan. Karena keterbatasan sumber daya dan waktu, penulis belum dapat membuktikan kemunculan Rare Pack.

REFERENSI

- [1] Bellevue, Wash. 2024. *Pokémon Trading Card Game Pocket Celebrates 60 Million Downloads Since Launching in October 2024* diakses pada 19 Juni 2025 dari <https://press.pokemon.com/en/Pokemon-Trading-Card-Game-Pocket-Celebrates-60-Million-Downloads-Since>,
- [2] Munir, Rinaldi. 2024. Teori Bilangan Bagian 1 diakses pada 19 Juni 2025 dari <https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/15-Teori-Bilangan-Bagian1-2024.pdf>
- [3] Munir, Rinaldi. 2024. Teori Bilangan Bagian 3 diakses pada 19 Juni 2025 dari <https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/17-Teori-Bilangan-Bagian3-2024.pdf>
- [4] Munir, Rinaldi. 2024. Kombinatorika Bagian 1 diakses pada 20 Juni 2025 dari <https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/18-Kombinatorika-Bagian1-2024.pdf>

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, 1 Juni 2025

A handwritten signature in black ink, appearing to read 'Jordan', with a stylized flourish at the end.

Muhammad Jordan Ferimeison 13524047