Jurnal Riset Mahasiswa Matematika
Volume 2(3) (2023), Pages 122-131
p-ISSN: 2808-1552; e-ISSN: 2808-4926

Pengamanan Pesan Menggunakan Algoritma One Time Pad dengan Linear Congruential Generator sebagai Pembangkit Kunci

Jamilatul Maghfiroh*, Turmudi, Elly Susanti

Program Studi Matematika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang

jamila.maghfiro@gmail.com*, turmudi_msi@mat.uin-malang.ac.id, ellysusanti@mat.uin-malang.ac.id

Abstrak

Kriptografi merupakan salah satu metode yang dapat digunakan untuk mengamankan suatu pesan dengan cara menyembunyikan teks aslinya. Penelitian ini menggunakan algoritma *One Time Pad* (OTP) untuk mengamankan sebuah pesan dan algoritma Linear *Congruential Generator* (LCG) sebagai pembangkit kunci. Tujuan dari penelitian ini adalah untuk mendeskripsikan proses pembangkitan kunci menggunakan algoritma LCG, proses enkripsi dan dekripsi pesan dengan menggunakan algoritma OTP. Adapun proses enkripsi dan dekripsi pesan dengan menggunakan algoritma OTP membutuhkan kunci sepanjang pesan aslinya dan harus acak. Sebab itu, dilakukan proses pembangkitan bilangan acak menggunakan algoritma LCG sebelum penyandian pesan. Hasil dari penelitian ini adalah algoritma LCG mampu membangun kunci yang dinamis dengan syarat panjang periodenya harus lebih besar atau sama dengan panjang *plaintext*nya. Proses penyandian pesan menggunakan algoritma OTP memiliki tingkat keamanan yang tinggi sebab jumlah karakter yang digunakan lebih banyak dan proses pengiriman pesan lebih mudah. *Ciphertext* yang dihasilkan merupakan pesan yang sangat acak dan tidak terbaca sehingga sulit dipecahkan.

Kata kunci: Bilangan Acak; Enkripsi; Dekripsi; Algoritma LCG; Algoritma OTP

Abstract

Cryptography is one method that can be used to secure a message by hiding the original text. This study uses the *One Time Pad* (OTP) algorithm to secure a message and the Linear *Congruential Generator* (LCG) algorithm as a key generator. The purpose of this study is to describe the key generation process using the LCG algorithm, the encryption and decryption process of messages using the OTP algorithm. The process of encrypting and decrypting messages using the OTP algorithm requires a key as long as the original message and must be random. Therefore, a random number generation process is carried out using the LCG algorithm before encoding the message. The results of this study are the LCG algorithm is able to build dynamic keys with the condition that the length of the period must be greater than or equal to the length of the *plaintext*. The process of encoding messages using the OTP algorithm has a high level of security because the number of characters used is more and the process of sending messages is easier. The resulting *ciphertext* is a very random and unreadable message that is difficult to decipher.

Keywords: Random Number; Encryption; Decryption; LCG Algorithm; OTP Algorithm.

PENDAHULUAN

Informasi merupakan salah satu aspek penting dalam kehidupan manusia. Informasi bisa bersifat rahasia atau umum. Adapun informasi yang bersifat rahasia sangat penting untuk dijaga keamanan kerahasiaannya. Kriptografi merupakan salah satu metode yang dapat digunakan untuk mengamankan suatu pesan dengan cara menyembunyikan pesan aslinya sehingga hanya mereka yang dimaksudkan yang dapat membaca dan memprosesnya. Pesan atau teks asli sebelum dilakukan proses apa pun disebut *plaintext*. Proses mengubah *plaintext* menjadi bentuk rahasia disebut enkripsi. Setelah teks asli dienkripsi, teks yang dihasilkan dikenal sebagai *ciphertext*. Proses pengubahan *ciphertext* menjadi *plaintext* dikenal sebagai dekripsi [1].

Submitted: 27 Juni 2022 Reviewed: 20 Januari 2023 Accepted: 28 Februari 2023
Doi: http://dx.doi.org/10.18860/jrmm.v2i3.16770

Salah satu algoritma klasik yang ada dalam kriptografi adalah algoritma *One Time Pad* (OTP). OTP adalah algoritma yang menggunakan kunci sepanjang *plaintext*-nya dan kunci yang digunakan benar-benar acak sehingga menghasilkan *ciphertext* yang juga benar-benar acak [2], oleh karena itu dibutuhkan suatu pembangkit bilangan acak agar kunci yang dihasilkan berupa barisan bilangan yang benar-benar acak. Salah satu algoritma yang dapat menghasilkan bilangan acak adalah Linear *Congruential Generator* (LCG). Algoritma ini digunakan untuk menghindari pembuatan kunci berulang. Dalam proses pembangkitan kunci, semakin panjang periodenya maka semakin kecil kemungkinan kunci tersebut diulang sehingga menghasilkan kunci yang dinamis [3].

Pada penelitian ini algoritma yang digunakan peneliti untuk mengamankan sebuah pesan rahasia adalah algoritma OTP. OTP digunakan karena merupakan algoritma sederhana namun kuat dengan tingkat keamanan yang tinggi sehingga tidak memungkinkan untuk dipecahkan dengan metode kriptanalitik apa pun [2]. Sedangkan LCG, algoritma tersebut digunakan karena memiliki kecepatan, kemudahan implementasi, dan sedikit menggunakan operasi bit.

KAJIAN PUSTAKA

Teorema 2.1.1

Jika a, b, c dan m bilangan bulat sedemikian sehingga $m > 0$, $d = gcd(c, m)$, dan $ac \equiv bc \ (mod\ m)$, maka $a \equiv b \ (mod \frac{m}{d})$.

Teorema 2.1.2 (Teorema Fundamental Aritmetika)

Setiap bilangan bulat positif $n > 1$ selalu dapat disajikan dalam bentuk perkalian bilangan-bilangan prima berpangkat. Representasi ini tunggal terhadap urutan factor-faktornya yaitu

$$n = {p_1}^{\alpha_1} {p_2}^{\alpha_2} \dots {p_k}^{\alpha_k}$$

Di mana $p_1 p_2 \dots p_k$ bilangan prima yang berbeda dan $\alpha_1 \alpha_2 \dots \alpha_k$ bilangan bulat positif.

Teorema 2.1.3 (Teorema Binomial)

Misalkan x dan y menjadi variable dan n adalah bilangan bulat positif. Maka,

$$(x+y)^n = \binom{n}{0} x^n + \binom{n}{1} x^{n-1} y + \binom{n}{2} x^{n-2} y^2 + \cdots + \binom{n}{n-2} x^2 y^{n-2} + \binom{n}{n-1} x y^{n-1} + \binom{n}{n} y^n$$

Atau menggunakan notasi penjumlahan

$$(x+y)^n = \sum_{j=0}^{n} \binom{n}{j} x^{n-j} y^j.$$

Teorema 2.1.4 (*Fermat's Little Theorem*)

Jika p prima dan a bilangan bulat positif, nilai $a^p - a$ adalah kelipatan p dan dinotasikan dengan

$$a^p \equiv a \ (mod\ p)$$

Jika a tidak habis dibagi oleh p yang berarti $(a, p) = 1$, maka sama dengan $a^{p-1} - 1$ adalah kelipatan p dan dinotasikan dengan $a^{p-1} \equiv 1 \ (mod\ p)$ [4].

Lemma 2.3.1

Misalkan $\{x_i\}$ menjadi barisan kongruen linier yang ditentukan oleh x_0, a, c, dan m didefinisikan sebagai berikut

$$x_{i+1} \equiv a x_i + c \ (mod\ m), \quad 0 \le x_{i+1} < m \qquad (1.1)$$

Asumsikan $a \ge 2$ maka

$$x_{i+k} \equiv a^k x_i + \frac{(a^k - 1)c}{a - 1} (mod\ m), \forall k \ge 0 \qquad (1.2)$$

Faktanya, suku pada barisan $\{x_i\}$ diberikan sebagai berikut

$$x_k \equiv a^k x_0 + \frac{(a^k - 1)c}{a - 1} (mod\ m), \forall k \ge 0 \qquad (1.3)$$

Lemma 2.3.2

Misalkan $m = {p_1}^{\alpha_1}{p_2}^{\alpha_2} \dots {p_n}^{\alpha_n}$ menjadi dekomposisi dari modulus m pangkat prima dimana $p_1 \dots p_n$ adalah bilangan prima yang berbeda dan semua $\alpha_j \in \mathbb{Z}^+$. Panjang periode terkecil adalah d dari sembarang barisan kongruensi linier $\{x_i\}$ yang ditentukan oleh $x_0, a, c,$ dan m yang mana $d = lcm(d_j)$ (panjang periode terkecil dari barisan kongruensi linier $\{x_i\}$ yang ditentukan oleh $x_0, a, c,$ dan ${p_j}^{\alpha_j}, 1 \le j \le n$).

Lemma 2.3.3

Misalkan p prima dan $\alpha \in \mathbb{Z}^+$ sedemikian sehingga $p^\alpha > 2$. Jika

$$x \equiv 1\ (mod\ p^\alpha), x \not\equiv 1 (mod\ p^{\alpha+1})$$

Maka

$$x^p \equiv 1\ (mod\ p^{\alpha+1}), x^p \not\equiv 1 (mod\ p^{\alpha+2})$$

Lemma 2.3.4

Jika $a \equiv 3 (mod\ 4)$ maka $\frac{a^{2^{\alpha-1}}-1}{a-1} \equiv 0 (mod\ 2^\alpha)$ ketika $\alpha > 1$ [5].

HASIL DAN PEMBAHASAN

1. Proses Pembangkitan Kunci dengan *Linear Congruential Generator* (LCG)

Adapun proses membangkitkan kunci dengan LCG sebagai berikut:

1. Menentukan *plaintext* atau pesan asli yang akan disandikan.
2. Menentukan nilai parameter yang akan digunakan, nilai a haruslah $0 < a < m$, nilai c haruslah $0 \le b < m$, nilai X_0 haruslah $0 \le X_0 < m$, dan nilai m haruslah $m > 0,\ m > a,\ m > c,\ m > X_0$.
3. Pengambilan nilai parameter pada $c \neq 0$ harus mengikuti Teorema 4.1.

   Teorema 4.1

   Linear congruential generators yang dibangun oleh barisan $x_i \equiv a^i x_0 + \frac{(a^i-1)c}{a-1} (mod\ m), \forall i \ge 0$, memiliki panjang periode sebesar nilai m jika memenuhi kondisi sebagai berikut:

   1) Parameter c dan m relatif prima,
   2) $a \equiv 1\ (mod\ p)$ jika p adalah faktor prima dari m,
   3) $a \equiv 1\ (mod\ 4)$ jika 4 habis membagi m (Knuth, 1981).

   Bukti:

   Dengan menggunakan Lemma 2.2, kita hanya perlu mempertimbangkan kasus modulus $m = p^\alpha$, dimana p prima dan $\alpha \in \mathbb{Z}^+$, karena ${p_1}^{\alpha_1}{p_2}^{\alpha_2} \dots {p_n}^{\alpha_n} = d = lcm(d_1, d_2, \dots, d_n) \le d_1, d_2, \dots, d_n \le {p_1}^{\alpha_1}{p_2}^{\alpha_2} \dots {p_n}^{\alpha_n}$ berlaku jika dan hanya jika $d_j = {p_j}^{\alpha_j}$ untuk $1 \le j \le n$.

   Untuk $a = 1$, misalkan $i = \frac{(a^i-1)}{a-1}$ maka $x_i \equiv x_0 + ic\ (mod\ m), \forall i \ge 0$ dan $x_i = x_0$ ketika $ic\ (mod\ m) = 0$ atau dapat dituliskan $ic \equiv 0\ (mod\ m)$ yaitu jika dan hanya jika $i \equiv 0 \left(mod \frac{m}{k}\right)$ dimana $k = gcd(c, m)$ (menurut Teorema 2.1.1). Jadi, $x_i = x_0$ jika dan hanya jika i adalah kelipatan dari $\frac{m}{k}$. Hal ini menyiratkan bahwa $\frac{m}{k}$ adalah panjang periode barisan $\{x_i\}$ dan memiliki panjang periode penuh sebesar m jika dan hanya jika $k = gcd(c, m) = 1$. Selanjutnya, karena $a = 1$ kita juga punya $1 \equiv 1\ (mod\ p)$ jika p adalah faktor prima dari m dan $1 \equiv 1\ (mod\ 4)$ jika 4 habis membagi m. Jadi, ketiga kondisi tersebut berlaku untuk $a = 1$.

   Untuk $a > 1$, perhatikan bahwa panjang periodenya adalah m jika dan hanya jika setiap bilangan bulat $0,1,2, \dots, (m-1)$ yang mungkin muncul dalam periode, tidak ada nilai yang muncul lebih dari satu kali. Oleh karena itu, panjang periode sama dengan m jika dan hanya jika $x_0 = 0$. Jadi, dapat diasumsikan bahwa panjang periode barisan $\{x_i\}$ adalah $m = p^\alpha$ dengan nilai awal $x_0 = 0$. Dengan mengunakan Lemma 2.3.1 maka barisan $\{x_i\}$ diberikan sebagai berikut

$$x_i \equiv \frac{(a^i - 1)c}{a-1} (mod\ m), \forall i \geq 1 \quad (2.1)$$

Ambil $x_i = 1$ maka diperoleh $\frac{(a^i-1)c}{a-1} \equiv 1\ (mod\ m)$. Selain itu, kongruensi ini berlaku jika dan hanya jika $gcd(c, m) = 1$ karena c memiliki invers modulo m, yaitu $\frac{(a^i-1)}{a-1} = 1 + a + a^2 + \cdots + a^{i-1} \in \mathbb{Z}$ (menurut deret geometri). Jadi, kondisi pertama telah dibuktikan [5].

Untuk membuktikan kondisi kedua dan ketiga, perhatikan fakta berikut: Panjang periode barisan adalah $m = p^\alpha$ jika dan hanya jika n adalah bilangan bulat positif terkecil sehingga $x_n = x_0 = 0$ yakni $n = m$. Ini menyiratkan bahwa $\frac{(a^n-1)c}{a-1} \equiv 0\ (mod\ m)$. Karena $gcd(c, m) = 1$ maka $\frac{(a^n-1)}{a-1} \equiv 0\ (mod\ m)$ (menurut Teorema 2.1.1). Untuk membuktikan bahwa $n = p^\alpha$, fakta diatas direduksi menjadi pembuktian Lemma berikut:

Lemma 4.1 Asumsikan bahwa $1 < a < p^\alpha$, dimana p prima. Jika n adalah bilangan bulat positif terkecil maka

$$n = p^\alpha \text{ jika dan hanya jika } \begin{cases} a \equiv 1\ (mod\ p), & p > 2 \\ a \equiv 1\ (mod\ 4), & p = 2 \end{cases}$$

Bukti:

($\Longrightarrow$)

Asumsikan bahwa $n = p^\alpha$ dan $p > 2$. Jika $a \not\equiv 1\ (mod\ p)$ maka $\frac{(a^n-1)}{a-1} \equiv 0\ (mod\ p^\alpha)$ jika dan hanya jika $a^n - 1 \equiv 0\ (mod\ p^\alpha)$. Karena $n = p^\alpha$ maka $a^{p^\alpha} - 1 \equiv 0\ (mod\ p^\alpha)$, artinya $a^{p^\alpha} - 1 = tp^\alpha$ untuk beberapa $t \in \mathbb{Z}^+$ atau dapat dinyatakan $a^{p^\alpha} - 1 = (tp^{\alpha-1})p$ dimana $tp^{\alpha-1} \in \mathbb{Z}^+$ dan p habis membagi $(a^{p^\alpha} - 1)$ sehingga $a^{p^\alpha} \equiv 1\ (mod\ p)$. Tetapi, menurut Teorema 2.1.4 menyatakan $a^{p^\alpha} \equiv a\ (mod\ p)$ jadi $a^n \equiv a\ (mod\ p)$. Hal ini berkontradiksi, oleh karena itu $a \equiv 1\ (mod\ p)$ [6].

Misalkan $p = 2$ sehingga $n = m = 2^\alpha, \alpha \geq 2$. Jika $a \equiv 1\ (mod\ 2)$ tetapi $a \not\equiv 1\ (mod\ 4)$, $a \equiv 3\ (mod\ 4)$ maka $\frac{a^{2^{\alpha-1}}-1}{a-1} \equiv 0\ (mod\ 2^\alpha)$ menurut Lemma 2.4. Hal ini kontradiksi karena $\frac{a^{2^\alpha}-1}{a-1} \equiv 0\ (mod\ 2^\alpha)$ ketika $n = m = 2^\alpha$, dimana $a \equiv 1\ (mod\ 4)$. Alternatifnya, gunakan $x_i \equiv ax_{i-1} + c\ (mod\ m)$, maka diperoleh

$$\begin{aligned} x_i &\equiv a(ax_{i-2} + c) + c\ (mod\ m) \\ &\equiv a^2 x_{i-2} + (a+1)c\ (mod\ m) \end{aligned}$$

Jadi, karena $x_0 = 0$ maka $x_2, x_4, x_6, \ldots$ adalah kelipatan dari $(a+1)c \equiv 4c\ (mod\ m)$. Akibatnya, barisan $\{x_{2i}\}$ memiliki panjang periode paling banyak $\frac{m}{4}$ sehingga pada $\{x_i\}$ memiliki panjang periode paling banyak $\frac{m}{2}$. Hal ini kontradiksi, oleh karena itu $a \equiv 1\ (mod\ 4)$ [5].

($\Longleftarrow$)

Asumsikan $n = p^\alpha$ dan amati bahwa kondisi kedua dan ketiga menyiratkan bahwa $a = 1 + qp^\beta$, dimana $p^\beta > 2$ dan q bukan kelipatan dari p. Dengan menggunakan Lemma 2.3.3, $a^p \equiv 1\ (mod\ p^{\beta+1})$ tetapi $a^p \not\equiv 1\ (mod\ p^{\beta+2})$. Dengan menggunakan induksi matematika pada γ, maka $a^{p^\gamma} \equiv 1\ (mod\ p^{\beta+\gamma})$ tetapi $a^{p^\gamma} \not\equiv 1\ (mod\ p^{\beta+\gamma+1}), \forall \gamma \geq 0$. Oleh karena itu,

$$\frac{a^{p^\gamma} - 1}{a-1} \equiv 0\ (mod\ p^\gamma);\ \frac{a^{p^\gamma} - 1}{a-1} \not\equiv 0\ (mod\ p^{\gamma+1}) \quad (2.2)$$

Persamaan (2.2) mengimplikasikan $\frac{a^{p^\alpha}-1}{a-1} \equiv 0\ (mod\ p^\alpha)$. Dengan menggunakan persamaan (2.1), barisan konruensi linier $\{x_i\}$ yang ditentukan oleh $0, a, c$, dan p^α sedemikian sehingga $x_i \equiv \frac{(a^i-1)c}{a-1} (mod\ p^\alpha), \forall i \geq 0$. Misalkan barisan ini memiliki panjang periode n, maka $x_i = x_0 = 0$ jika dan hanya jika i adalah kelipatan dari n, yaitu $\frac{a^i-1}{a-1} \equiv$

$0\ (mod\ p^{\alpha})$ karena $(c, m) = 1$ (menurut Teorema 2.1.1). Jadi, p^{α} adalah kelipatan n, yang hanya dapat terjadi jika $n = p^{\gamma}$, untuk suatu bilangan bulat $\gamma \geq 0$ dan persamaan (2.2) mengimplikasikan $n = p^{\alpha}$. Jadi, kondisi kedua dan ketiga telah terbukti [6].

4. Melakukan perhitungan LCG dengan rumus:
$$X_n = (a * X_{n-i} + c)\ (mod\ m)$$
5. Panjang periode harus lebih besar atau sama dengan panjang plainteks.

Adapun implementasinya adalah sebagai berikut:

Si A akan mengirimkan sebuah pesan rahasia kepada si B, agar kunci tak rekursif maka panjang periode barisan bilangan acak tersebut tidak boleh kurang dari panjang *plaintext*nya. *Plaintext* yang akan disandikan adalah "Satgas COVID-19" dengan panjang *plaintext* sebesar 15. Si A melakukan beberapa kali percobaan dalam membangkitkan bilangan acak dengan menggunakan parameter yang berbeda-beda.

Percobaan pertama :

Nilai parameter yang digunakan oleh si A adalah $c \neq 0$ dan $a \neq 1$, dipilih $c = 18$, $m = 38$, $a = 7$, dan $(X_0) = 33$. Parameter yang dipilih tidak memenuhi kondisi pertama dan didapatkan hasil perhitungan nilai LCG sebagai berikut:

$$X_1 = (7 \cdot X_0 + 18)\ (mod\ 38) = (7 \cdot 33 + 18)(mod\ 38) = 21$$
$$X_2 = (7 \cdot X_1 + 18)\ (mod\ 38) = (7 \cdot 21 + 18)(mod\ 38) = 13$$
$$X_3 = (7 \cdot X_2 + 18)\ (mod\ 38) = (7 \cdot 13 + 18)(mod\ 38) = 33$$
$$X_4 = (7 \cdot X_3 + 18)\ (mod\ 38) = (7 \cdot 33 + 18)(mod\ 38) = 21$$
$$X_5 = (7 \cdot X_4 + 18)\ (mod\ 38) = (7 \cdot 21 + 18)(mod\ 38) = 13$$

Setelah dilakukan perhitungan dapat dilihat bahwa panjang periodenya 3, kurang dari m juga kurang 15.

Percobaan kedua :

Nilai parameter yang digunakan oleh si A adalah $c \neq 0$ dan $a \neq 1$, dipilih $c = 17$, $m = 60$, $a = 45$, dan $(X_0) = 33$. Parameter yang dipilih tidak memenuhi kondisi kedua dan didapatkan hasil perhitungan nilai LCG sebagai berikut:

$$X_1 = (45 \cdot X_0 + 17)\ (mod\ 60) = (45 \cdot 33 + 17)(mod\ 60) = 2$$
$$X_2 = (45 \cdot X_1 + 17)\ (mod\ 60) = (45 \cdot 2 + 17)(mod\ 60) = 47$$
$$X_3 = (45 \cdot X_2 + 17)\ (mod\ 60) = (45 \cdot 47 + 17)(mod\ 60) = 32$$
$$X_4 = (45 \cdot X_3 + 17)\ (mod\ 60) = (45 \cdot 32 + 17)(mod\ 60) = 17$$
$$X_5 = (45 \cdot X_4 + 17)\ (mod\ 60) = (45 \cdot 17 + 17)(mod\ 60) = 2$$
$$X_6 = (45 \cdot X_5 + 17)\ (mod\ 60) = (45 \cdot 2 + 17)(mod\ 60) = 47$$

Setelah dilakukan perhitungan dapat dilihat bahwa panjang periodenya 4, kurang dari m juga kurang dari 15.

Percobaan ketiga :

Nilai parameter yang digunakan oleh si A adalah $c \neq 0$ dan $a \neq 1$, dipilih $c = 17$, $m = 40$, $a = 31$, dan $(X_0) = 33$. Parameter yang dipilih tidak memenuhi kondisi ketiga dan didapatkan hasil perhitungan nilai LCG sebagai berikut:

$$X_1 = (31 \cdot X_0 + 17)\ (mod\ 40) = (31 \cdot 33 + 17)(mod\ 40) = 0$$
$$X_2 = (31 \cdot X_1 + 17)\ (mod\ 40) = (31 \cdot 0 + 17)(mod\ 40) = 17$$
$$X_3 = (31 \cdot X_2 + 17)\ (mod\ 40) = (31 \cdot 17 + 17)(mod\ 40) = 24$$
$$X_4 = (31 \cdot X_3 + 17)\ (mod\ 40) = (31 \cdot 24 + 17)(mod\ 40) = 1$$
$$X_5 = (31 \cdot X_4 + 17)\ (mod\ 40) = (31 \cdot 1 + 17)(mod\ 40) = 8$$
$$X_6 = (31 \cdot X_5 + 17)\ (mod\ 40) = (31 \cdot 8 + 17)(mod\ 40) = 25$$
$$X_7 = (31 \cdot X_6 + 17)\ (mod\ 40) = (31 \cdot 25 + 17)(mod\ 40) = 32$$
$$X_8 = (31 \cdot X_7 + 17)\ (mod\ 40) = (31 \cdot 32 + 17)(mod\ 40) = 9$$
$$X_9 = (31 \cdot X_8 + 17)\ (mod\ 40) = (31 \cdot 9 + 17)(mod\ 40) = 16$$
$$X_{10} = (31 \cdot X_9 + 17)\ (mod\ 40) = (31 \cdot 16 + 17)(mod\ 40) = 33$$
$$X_{11} = (31 \cdot X_{10} + 17)\ (mod\ 40) = (31 \cdot 33 + 17)(mod\ 40) = 0$$
$$X_{12} = (31 \cdot X_{11} + 17)\ (mod\ 40) = (31 \cdot 0 + 17)(mod\ 40) = 17$$

Setelah dilakukan perhitungan dapat dilihat bahwa panjang periodenya 10, kurang dari m juga kurang dari 15.

Percobaan keempat :

Nilai parameter yang digunakan oleh si A adalah $c \neq 0$ dan $a \neq 1$, dipilih $c = 17$, $m = 50$, $a = 11$, dan $(X_0) = 33$. Parameter yang dipilih memenuhi ketiga kondisi dan didapatkan hasil perhitungan nilai LCG sebagai berikut:

$$X_1 = (11 \cdot X_0 + 17)\ (mod\ 50) = (11 \cdot 33 + 17)(mod\ 50) = 30$$
$$X_2 = (11 \cdot X_1 + 17)\ (mod\ 50) = (11 \cdot 30 + 17)(mod\ 50) = 47$$
$$X_3 = (11 \cdot X_2 + 17)\ (mod\ 50) = (11 \cdot 47 + 17)(mod\ 50) = 34$$
$$X_4 = (11 \cdot X_3 + 17)\ (mod\ 50) = (11 \cdot 34 + 17)(mod\ 50) = 41$$
$$X_5 = (11 \cdot X_4 + 17)\ (mod\ 50) = (11 \cdot 41 + 17)(mod\ 50) = 18$$
$$X_6 = (11 \cdot X_5 + 17)\ (mod\ 50) = (11 \cdot 18 + 17)(mod\ 50) = 15$$
$$X_7 = (11 \cdot X_6 + 17)\ (mod\ 50) = (11 \cdot 15 + 17)(mod\ 50) = 32$$
$$X_8 = (11 \cdot X_7 + 17)\ (mod\ 50) = (11 \cdot 32 + 17)(mod\ 50) = 19$$
$$X_9 = (11 \cdot X_8 + 17)\ (mod\ 50) = (11 \cdot 19 + 17)(mod\ 50) = 26$$
$$X_{10} = (11 \cdot X_9 + 17)\ (mod\ 50) = (11 \cdot 26 + 17)(mod\ 50) = 3$$
$$X_{11} = (11 \cdot X_{10} + 17)\ (mod\ 50) = (11 \cdot 3 + 17)(mod\ 50) = 0$$
$$X_{12} = (11 \cdot X_{11} + 17)\ (mod\ 50) = (11 \cdot 0 + 17)(mod\ 50) = 17$$
$$X_{13} = (11 \cdot X_{12} + 17)\ (mod\ 50) = (11 \cdot 17 + 17)(mod\ 50) = 4$$
$$X_{14} = (11 \cdot X_{13} + 17)\ (mod\ 50) = (11 \cdot 4 + 17)(mod\ 50) = 11$$
$$X_{15} = (11 \cdot X_{14} + 17)\ (mod\ 50) = (11 \cdot 11 + 17)(mod\ 50) = 38$$
$$X_{16} = (11 \cdot X_{15} + 17)\ (mod\ 50) = (11 \cdot 38 + 17)(mod\ 50) = 35$$
$$X_{17} = (11 \cdot X_{16} + 17)\ (mod\ 50) = (11 \cdot 35 + 17)(mod\ 50) = 2$$
$$X_{18} = (11 \cdot X_{17} + 17)\ (mod\ 50) = (11 \cdot 2 + 17)(mod\ 50) = 39$$
$$X_{19} = (11 \cdot X_{18} + 17)\ (mod\ 50) = (11 \cdot 39 + 17)(mod\ 50) = 46$$
$$X_{20} = (11 \cdot X_{19} + 17)\ (mod\ 50) = (11 \cdot 46 + 17)(mod\ 50) = 23$$
$$X_{21} = (11 \cdot X_{20} + 17)\ (mod\ 50) = (11 \cdot 23 + 17)(mod\ 50) = 20$$
$$X_{22} = (11 \cdot X_{21} + 17)\ (mod\ 50) = (11 \cdot 20 + 17)(mod\ 50) = 37$$
$$X_{23} = (11 \cdot X_{22} + 17)\ (mod\ 50) = (11 \cdot 37 + 17)(mod\ 50) = 24$$
$$X_{24} = (11 \cdot X_{23} + 17)\ (mod\ 50) = (11 \cdot 24 + 17)(mod\ 50) = 31$$
$$X_{25} = (11 \cdot X_{24} + 17)\ (mod\ 50) = (11 \cdot 31 + 17)(mod\ 50) = 8$$
$$X_{26} = (11 \cdot X_{25} + 17)\ (mod\ 50) = (11 \cdot 8 + 17)(mod\ 50) = 5$$
$$X_{27} = (11 \cdot X_{26} + 17)\ (mod\ 50) = (11 \cdot 5 + 17)(mod\ 50) = 22$$
$$X_{28} = (11 \cdot X_{27} + 17)\ (mod\ 50) = (11 \cdot 22 + 17)(mod\ 50) = 9$$
$$X_{29} = (11 \cdot X_{28} + 17)\ (mod\ 50) = (11 \cdot 9 + 17)(mod\ 50) = 16$$
$$X_{30} = (11 \cdot X_{29} + 17)\ (mod\ 50) = (11 \cdot 16 + 17)(mod\ 50) = 43$$
$$X_{31} = (11 \cdot X_{30} + 17)\ (mod\ 50) = (11 \cdot 43 + 17)(mod\ 50) = 40$$
$$X_{32} = (11 \cdot X_{31} + 17)\ (mod\ 50) = (11 \cdot 40 + 17)(mod\ 50) = 7$$
$$X_{33} = (11 \cdot X_{32} + 17)\ (mod\ 50) = (11 \cdot 7 + 17)(mod\ 50) = 44$$
$$X_{34} = (11 \cdot X_{33} + 17)\ (mod\ 50) = (11 \cdot 44 + 17)(mod\ 50) = 1$$
$$X_{35} = (11 \cdot X_{34} + 17)\ (mod\ 50) = (11 \cdot 1 + 17)(mod\ 50) = 28$$
$$X_{36} = (11 \cdot X_{35} + 17)\ (mod\ 50) = (11 \cdot 28 + 17)(mod\ 50) = 25$$
$$X_{37} = (11 \cdot X_{36} + 17)\ (mod\ 50) = (11 \cdot 25 + 17)(mod\ 50) = 42$$
$$X_{38} = (11 \cdot X_{37} + 17)\ (mod\ 50) = (11 \cdot 42 + 17)(mod\ 50) = 29$$
$$X_{39} = (11 \cdot X_{38} + 17)\ (mod\ 50) = (11 \cdot 29 + 17)(mod\ 50) = 36$$
$$X_{40} = (11 \cdot X_{39} + 17)\ (mod\ 50) = (11 \cdot 36 + 17)(mod\ 50) = 13$$
$$X_{41} = (11 \cdot X_{40} + 17)\ (mod\ 50) = (11 \cdot 13 + 17)(mod\ 50) = 10$$
$$X_{42} = (11 \cdot X_{41} + 17)\ (mod\ 50) = (11 \cdot 10 + 17)(mod\ 50) = 27$$
$$X_{43} = (11 \cdot X_{42} + 17)\ (mod\ 50) = (11 \cdot 27 + 17)(mod\ 50) = 14$$
$$X_{44} = (11 \cdot X_{43} + 17)\ (mod\ 50) = (11 \cdot 14 + 17)(mod\ 50) = 21$$
$$X_{45} = (11 \cdot X_{44} + 17)\ (mod\ 50) = (11 \cdot 21 + 17)(mod\ 50) = 48$$
$$X_{46} = (11 \cdot X_{45} + 17)\ (mod\ 50) = (11 \cdot 48 + 17)(mod\ 50) = 45$$
$$X_{47} = (11 \cdot X_{46} + 17)\ (mod\ 50) = (11 \cdot 45 + 17)(mod\ 50) = 12$$
$$X_{48} = (11 \cdot X_{47} + 17)\ (mod\ 50) = (11 \cdot 12 + 17)(mod\ 50) = 49$$
$$X_{49} = (11 \cdot X_{48} + 17)\ (mod\ 50) = (11 \cdot 49 + 17)(mod\ 50) = 6$$
$$X_{50} = (11 \cdot X_{49} + 17)\ (mod\ 50) = (11 \cdot 6 + 17)(mod\ 50) = 33$$

$$X_{51} = (11 \cdot X_{50} + 17)\ (mod\ 50) = (11 \cdot 33 + 17)(mod\ 50) = 30$$
$$X_{52} = (11 \cdot X_{51} + 17)\ (mod\ 50) = (11 \cdot 30 + 17)(mod\ 50) = 47$$

Setelah dilakukan perhitungan dapat dilihat bahwa panjang periodenya sebesar m yaitu 50 dan lebih dari 15.

Setelah melakukan beberapa kali percobaan, si A memutuskan yang akan dijadikan sebagai kunci dalam penyandian pesan adalah barisan bilangan acak yang dihasilkan pada percobaan keempat, karena memiliki periode penuh. Panjang *plaintext* sebesar 15 maka kunci yang akan digunakan adalah 30 47 34 41 18 15 32 19 26 3 0 17 4 11 38.

2. Proses Enkripsi Pesan menggunakan Algoritma *One Time Pad* (OTP) dan *Linear Congruential Generator* (LCG) sebagai pembangkit kunci

Adapun proses enkripsi algoritma OTP sebagai berikut:

1. Menentukan *plaintext* atau pesan asli yang akan disandikan.
2. Mengonversikan *plaintext* ke nilai karakter menurut tabel ASCII.
3. Mengambil kunci yang telah dibangkitkan dari perhitungan LCG.
4. Melakukan proses enkripsi (penyandian) pesan dengan menggunakan algoritma OTP di mana nilai modulonya 256 (disesuaikan jumlah karakter pada tabel ASCII),
$$C_i = P_i + K_i(mod\ n)$$
5. Mengonversi nilai hasil enkripsi ke karakter menurut tabel ASCII.
6. Mendapatkan pesan berupa pesan tersandi (*ciphertext*).

Adapun implementasinya sebagai berikut:

1. Si A akan mengirimkan sebuah pesan rahasia kepada si B, *plaintext* atau pesan asli yang akan disandikan adalah "Satgas COVID-19".
2. *Plaintext* dikonversikan ke nilai desimal menurut tabel ASCII.

S = 83
a = 97
t = 116
g = 103
a = 97
s = 115
C = 67
O = 79
V = 86
I = 73
D = 68
−= 45
1 = 49
9 = 57

3. Pengambilan kunci yang akan digunakan dalam proses enkripsi pesan adalah barisan bilangan acak yang telah dibangkitkan dari perhitungan LCG yakni 30 47 34 41 18 15 32 19 26 3 0 17 4 11 38.
4. Melakukan proses enkripsi (penyandian) pesan dengan menggunakan algoritma OTP,

$$P_1 + K_1(mod\ 256) = 83 + 30(mod\ 256) = 113 = C_1$$
$$P_2 + K_2(mod\ 256) = 97 + 47(mod\ 256) = 144 = C_2$$
$$P_3 + K_3(mod\ 256) = 116 + 34(mod\ 256) = 150 = C_3$$
$$P_4 + K_4(mod\ 256) = 103 + 41(mod\ 256) = 144 = C_4$$
$$P_5 + K_5(mod\ 256) = 97 + 18(mod\ 256) = 115 = C_5$$
$$P_6 + K_6(mod\ 256) = 115 + 15(mod\ 256) = 130 = C_6$$
$$P_7 + K_7(mod\ 256) = 32 + 32(mod\ 256) = 64 = C_7$$
$$P_8 + K_8(mod\ 256) = 57 + 19(mod\ 256) = 76 = C_8$$
$$P_9 + K_9(mod\ 256) = 79 + 26(mod\ 256) = 105 = C_9$$
$$P_{10} + K_{10}(mod\ 256) = 8 + 3(mod\ 256) = 11 = C_{10}$$
$$P_{11} + K_{11}(mod\ 256) = 73 + 0(mod\ 256) = 73 = C_{11}$$

$$P_{12} + K_{12}(mod\ 256) = 68 + 17(mod\ 256) = 85 = C_{12}$$
$$P_{13} + K_{13}(mod\ 256) = 45 + 4(mod\ 256) = 49 = C_{13}$$
$$P_{14} + K_{14}(mod\ 256) = 49 + 11(mod\ 256) = 60 = C_{14}$$
$$P_{15} + K_{15}(mod\ 256) = 57 + 38(mod\ 256) = 95 = C_{15}$$

5. Mengonversi nilai hasil enkripsi ke karakter menurut tabel ASCII.

$$113 = q$$
$$144 = \text{É}$$
$$150 = \text{û}$$
$$144 = \text{É}$$
$$115 = s$$
$$130 = \text{é}$$
$$64 = @$$
$$76 = L$$
$$105 = i$$
$$11 = VT$$
$$73 = I$$
$$85 = U$$
$$49 = 1$$
$$60 = <$$
$$95 = \_$$

6. *Ciphertext* yang dihasilkan adalah qÉûÉsé@LiVTIU1<_.

Pesan qÉûÉsé@LiVTIU1<_ yang akan dikirimkan si A kepada si B beserta nilai parameter yang digunakan pada perhitungan LCG yakni $X_0 = 33$, $c = 17$, $m = 50$, dan $a = 11$.

3. Proses Dekripsi Pesan menggunakan Algoritma *One Time Pad* (OTP) dan *Linear Congruential Generator* (LCG) sebagai pembangkit kunci

Sebelum melakukan proses dekripsi si penerima harus melakukan perhitungan LCG terlebih dahulu untuk mendapatkan kunci karena kunci yang diberikan oleh pengirim hanya nilai parameternya saja. Adapun proses dekripsi algoritma OTP sebagai berikut:

1. Menerima *ciphertext* atau pesan tersandi dan nilai parameter yang digunakan pengirim untuk melakukan perhitungan LCG.
2. Melakukan perhitungan nilai LCG dengan menggunakan parameter yang sama seperti yang digunakan oleh pengirim.
3. Mengonversikan *ciphertext* ke nilai karakter menurut tabel ASCII.
4. Melakukan proses dekripsi pesan dengan menggunakan algoritma OTP

$$P_i = C_i - K_i(mod\ n)$$

Di mana nilai modulonya 256 (disesuaikan jumlah karakter pada tabel ASCII) dan menggunakan kunci yang telah dibangkitkan dari perhitungan LCG.

5. Mengonversi nilai hasil dekripsi ke karakter menurut tabel ASCII.
6. Mendapatkan pesan berupa pesan asli (*plaintext*).

Adapun implementasinya sebagai berikut:

1. Si B menerima sebuah pesan tersandi yang dikirimkan oleh si A, *ciphertext* atau pesan tersandi tersebut adalah qÉûÉsé@LiVTIU1<_.
2. Si B menerima nilai parameter yang dikirimkan oleh si A, nilai parameternya adalah $X_0 = 33$, $c = 17$, $m = 50$, dan $a = 11$.
3. Si B melakukan proses perhitungan LCG sampai X_{15} disesuaikan panjang *ciphertext*nya.
4. Hasil yang didapatkan oleh si B sama seperti hasil yang didapatkan oleh si A, yakni 30 47 34 41 18 15 32 19 26 3 0 17 4 11 38.

Adapun proses dekripsi algoritma OTP sebagai berikut:

1. *Ciphertext* atau pesan tersandi adalah qÉûÉsé@LiVTIU1<_.
2. Mengonversikan *ciphertext* ke nilai desimal menurut tabel ASCII

q = 113
É = 144
û = 150

É = 144
s = 115
é = 130
@ = 64
L = 76
i = 105
VT = 11
I = 73
U = 85
1 = 49
< = 60
_ = 95

3. Kunci yang akan digunakan adalah barisan bilangan acak yang dihasilkan dari perhitungan LCG yakni 30 47 34 41 18 15 32 19 26 3 0 17 4 11 38.
4. Melakukan proses dekripsi pesan dengan menggunakan algoritma OTP,

$$C_1 - K_1(mod\ 256) = 113 - 30(mod\ 256) = 83 = P_1$$
$$C_2 - K_2(mod\ 256) = 144 - 47(mod\ 256) = 97 = P_2$$
$$C_3 - K_3(mod\ 256) = 150 - 34(mod\ 256) = 116 = P_3$$
$$C_4 - K_4(mod\ 256) = 144 - 41(mod\ 256) = 103 = P_4$$
$$C_5 - K_5(mod\ 256) = 115 - 18(mod\ 256) = 97 = P_5$$
$$C_6 - K_6(mod\ 256) = 130 - 15(mod\ 256) = 115 = P_6$$
$$C_7 - K_7(mod\ 256) = 64 - 32(mod\ 256) = 32 = P_7$$
$$C_8 - K_8(mod\ 256) = 76 - 19(mod\ 256) = 57 = P_8$$
$$C_9 - K_9(mod\ 256) = 105 - 26(mod\ 256) = 79 = P_9$$
$$C_{10} - K_{10}(mod\ 256) = 11 - 3(mod\ 256) = 8 = P_{10}$$
$$C_{11} - K_{11}(mod\ 256) = 73 - 0(mod\ 256) = 73 = P_{11}$$
$$C_{12} - K_{12}(mod\ 256) = 85 - 17(mod\ 256) = 68 = P_{12}$$
$$C_{13} - K_{13}(mod\ 256) = 49 - 4(mod\ 256) = 45 = P_{13}$$
$$C_{14} - K_{14}(mod\ 256) = 60 - 11(mod\ 256) = 49 = P_{14}$$
$$C_{15} - K_{15}(mod\ 256) = 95 - 38(mod\ 256) = 57 = P_{15}$$

5. Mengonversi nilai hasil dekripsi ke karakter menurut tabel ASCII

$$83 = S$$
$$97 = a$$
$$116 = t$$
$$103 = g$$
$$97 = a$$
$$115 = s$$
$$32 = spasi$$
$$67 = C$$
$$79 = O$$
$$86 = V$$
$$73 = I$$
$$68 = D$$
$$45 = -$$
$$49 = 1$$
$$57 = 9$$

6. Si B mendapatkan *plaintext* atau teks asli berupa “Satgas COVID-19”. Pesan tersebut sesuai seperti teks asli milik si A.

KESIMPULAN

1. Algoritma LCG mampu membantu dalam memenuhi kebutuhan sebagai kunci dalam proses enkripsi dan dekripsi pesan pada algoritma OTP, sebab barisan bilangan yang dihasilkan oleh

algoritma ini mampu menghindari penggunaan kunci yang rekursif dengan syarat pemilihan parameternya mengikuti Teorema 4.1 dan panjang periodenya lebih besar atau sama dengan panjang *plaintext* nya.

2. Pada proses enkripsi, pesan dikonversikan ke dalam karakter pada tabel ASCII, sehingga tingkat keamanan penyandian pesan menjadi lebih tinggi sebab jumlah karakter yang digunakan lebih banyak. Selain itu, proses pengiriman pesan oleh pengirim menjadi lebih mudah sebab kunci yang harus dikirimkan tidak sepanjang teks aslinya melainkan hanya nilai parameternya saja.
3. Pesan tersandi (*ciphertext*) menjadi sulit terpecahkan sebab barisan kunci yang digunakan benar-benar acak sehingga bentuk pesan tersandi pun menjadi acak dan tak terbaca. Namun, jika kunci yang digunakan tidak sesuai dengan milik pengirim maka *plaintext* yang dihasilkan tidak sama dengan milik pengirim.

DAFTAR PUSTAKA

[1] Munir, M. Wasim. 2005. Cryptography. Pakistan: SZABIST

[2] Mollin, Richard A. 2007. An Introduction to Cryptography. London: Taylor & Francis Group.

[3] Stallings, William. 2005. Cryptography and Network Security Principles and Practices. Amerika: Pearson Education Inc.

[4] Rossen, Kenneth H. 2000. Elementary Number Theory. London: Addison-Wesley.

[5] Glen, Amy. 2002. On the Period Length of Pseudorandom Number Sequences. Australia: The University of Adelaide.

[6] Knuth, Donald E. 1981. The Art of Computer Programming. Amerika: Addison-Wesley.