Implementasi Fitur Enkripsi Gambar dengan Arnold's Cat Map pada Layanan Cloud Storage untuk Menambah Lapisan Keamanan

Muhammad Rakha Athaya - 13520108
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail (gmail): 13520108@std.stei.itb.ac.id

***Abstrak*—Layanan *Cloud Storage* menjadi salah satu teknologi penting dalam menghadapi kebutuhan penyimpanan data digital yang membesar. Berbagai fitur yang disediakan layanan *cloud storage* memberikan manfaat besar bagi pengguna. Jumlah data pengguna yang besar memberikan tuntutan keamanan yang ketat bagi teknologi ini. Namun, sebagian besar layanan *cloud storage* masih belum melindungi data pengguna dengan enkripsi *end-to-end* di mana data pengguna berkemungkinan untuk diakses oleh penyedia layanan. Lapisan keamanan dapat ditambahkan dari sisi pengguna dengan menyediakan fitur enkripsi dan dekripsi file. Untuk file berupa gambar, fungsi *chaos* seperti *Arnold's Cat Map* dapat digunakan untuk menjawab permasalahan tersebut.**

Keywords—Cloud Storage, Chaos, Arnold's Cat Map, Kriptografi, Enkripsi, Dekripsi.

I. PENDAHULUAN

Pada era digital ini berbagai inovasi hadir untuk memudahkan kehidupan sehari-hari. Seiring dengan adaptasi masyarakat terhadap perkembangan teknologi, kebutuhan akan ruang penyimpanan digital terus meningkat. Salah satu inovasi teknologi untuk menjawab kebutuhan tersebut adalah layanan cloud storage. Cloud storage merupakan teknologi penyimpanan data digital yang memungkinkan pengguna untuk menyimpan, mengakses, dan mengelola data miliknya melalui koneksi jaringan internet. Fitur-fitur dari layanan cloud storage umumnya meliputi aksesibilitas data dari mana saja, kapasitas penyimpanan yang fleksibel, serta kemampuan untuk berbagi dan sinkronisasi data dari berbagai perangkat dengan mudah. Dalam pemakaian sehari-hari, pengguna dapat menyimpan cadangan untuk file foto, video, dan dokumen penting di cloud storage. Hal ini memastikan data penting pengguna aman dari kehilangan. Selain itu, berbagai layanan cloud storage juga memfasilitasi kolaborasi kerja jarak jauh dengan memungkinkan banyak orang untuk berbagi dan mengedit dokumen bersama-sama.

Meskipun saat ini layanan cloud storage sudah memberikan lapisan keamanan dengan enkripsi data dalam pengiriman (*in transit*) dan dalam penyimpanan (*at rest*), sebagian besar masih tidak menerapkan enkripsi *end-to-end* dari awal bagi pengguna. Enkripsi *end-to-end* itu sendiri adalah metode enkripsi di mana data dienkripsi di perangkat pengguna dan hanya bisa didekripsi di perangkat yang disetujui pengguna. Hal ini berarti data pengguna dienkripsi sebelum diunggah ke cloud dan hanya bisa diakses kembali oleh pengguna yang memiliki kunci enkripsinya. Dengan kata lain, baik penyedia layanan cloud maupun pihak ketiga lainnya tidak memiliki kemampuan untuk mendekripsi dan mengakses data pengguna, bahkan jika mereka ingin melakukannya. Sebagai contoh, layanan cloud terbesar seperti Google Drive dan Dropbox melakukan enkripsi data pada saat *in transit* di jaringan dan *at rest* di server, akan tetapi mereka tetap memiliki kemampuan untuk mendekripsi dan mengakses data pengguna yang tersimpan tersebut. Keadaan ini menimbulkan bahaya dan potensi masalah terkait privasi dan keamanan data pengguna. Data yang disimpan di cloud storage masih rentan terhadap akses yang disebabkan oleh kebijakan penyedia layanan yang memungkinkan mereka untuk mengakses data pengguna dengan suatu alasan tertentu.

Contoh nyata dari potensi perubahan kebijakan tersebut adalah pada tanggal 5 Juni 2024, Adobe melakukan pembaruan terhadap Syarat dan Ketentuan Layanannya (ToS). Dalam pembaruan ini, banyak pengguna yang memprotes perubahan ketentuan yang mengizinkan Adobe untuk melakukan pemindaian terhadap file-file pengguna yang tersimpan pada Adobe Creative Cloud dengan tujuan meningkatkan layanannya. Penambahan klausa yang samar dan multitafsir ini membuat pengguna merasa bahwa hak privasi terhadap file-file pribadi yang disimpan di layanan cloud tersebut dilanggar. Kejadian serupa dapat makin sering terjadi dengan adanya kebutuhan data yang sangat besar karena berbagai perusahaan menunjukkan ketertarikan di bidang *machine learning* sekarang.

Oleh karena itu, ada kebutuhan untuk solusi keamanan tambahan yang dapat melindungi keamanan dan privasi data pengguna. Dalam makalah ini akan dibahas mengenai implementasi fitur enkripsi dan dekripsi file gambar agar dapat memberikan lapisan keamanan tambahan bagi pengguna. Dengan adanya fitur enkripsi dan dekripsi file gambar ini, pihak-pihak selain pengguna yang memiliki kunci rahasia tidak dapat mengakses data file gambar milik pengguna dan menggunakannya tanpa diberikan akses langsung oleh pengguna.

II. TEORI DASAR

A. Cloud Storage

Cloud storage merupakan sebuah model teknologi digital di mana data disimpan pada berbagai server yang diatur dan dijalankan oleh pihak penyedia layanan. Model ini memungkinkan pengguna untuk menyimpan dan mengakses datanya tanpa harus membeli perangkat keras baru ataupun menjalankan sebuah server sendiri. Pengguna dapat mengakses *cloud storage* lewar jaringan internet. Keuntungan utama dari teknologi ini adalah pengguna dapat menghemat biaya operasional serta memperoleh fleksibilitas yang tinggi.

Penyedia layanan cloud storage menjaga keamanan data para pengguna menggunakan berbagai metode kriptografi dan redudansi. Data pengguna biasanya dienkripsi pada saat berada dalam jaringan dan pada saat disimpan pada server. Hal ini meminimalisir kemungkinan terjadinya kebocoran informasi meskipun terdapat server yang mengalami gangguan atau serangan.

B. Teori Chaos

Teori *Chaos* adalah sebuah cabang dalam bidang matematika mengenai sistem-sistem non-linear dan dinamis yang memiliki sifat yang *chaotic* (tidak terduga) dan juga sistem-sistem yang tampak tak terduga namun tetap memiliki keteraturan di dalamnya. Sistem-sistem ini sangat sensitif terhadap parameter dan nilai awal, di mana perubahan nilai sangat kecil di awal dapat menimbulkan efek yang besar pada hasil akhirnya.

C. Arnold's Cat Map

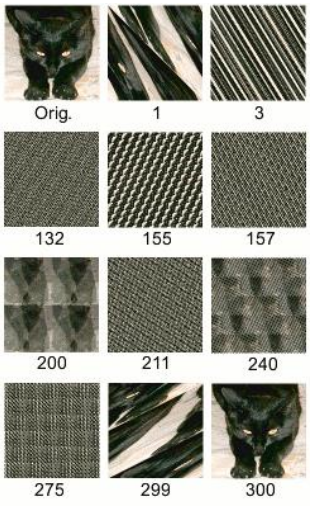
Arnold's Cat Map (ACM) adalah salah satu fungsi *chaos* sederhana yang diperkenalkan oleh Vladimir Arnold. Fungsi ini menghasilkan transformasi peregangan pada sebuah bidang dua dimensi yang kemudian disusun kembali dalam area yang tetap. Pemanfaatan utama dari transformasi ini adalah untuk mengacak posisi dari setiap titik dalam gambar. *Arnold's Cat Map* didefinisikan dengan persamaan berikut.

$$\begin{bmatrix} x_{i+1} \\ y_{i+1} \end{bmatrix} = \begin{bmatrix} 1 & b \\ c & bc+1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix} mod\ N$$

Dalam persamaan di atas, x_i dan y_i adalah koordinat semula dari sebuah titik pada gambar. Titik tersebut kemudian akan dipindahkan ke posisi x_{i+1} dan y_{i+1} berdasarkan transformasi dengan matriks berdeterminan 1 di atas. Operasi modulo dengan N (ukuran bidang gambar) memastikan setiap titik dipetakan dalam ruang lingkup gambar aslinya. Hal ini juga menyebabkan ukuran bidang gambar harus berbentuk persegi ($N \times N$).

Untuk proses dekripsi, digunakan persamaan matriks *inverse* dari *Arnold's Cat Map* sebagai berikut.

$$\begin{bmatrix} x_{i+1} \\ y_{i+1} \end{bmatrix} = \begin{bmatrix} bc+1 & -b \\ -c & 1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix} mod\ N$$



Gambar 1 Gambar Asli Akan Kembali Pada Iterasi Tertentu

Meskipun hasil dari *Arnold's Cat Map* memiliki sifat yang *chaotic* , gambar yang diacak dapat membentuk gambar asli kembali setelah beberapa melalu beberapa iterasi tertentu. Hal tersebut dipengaruhi oleh nilai dalam matriks beserta ukuran dari gambar yang diacak (N).

III. IMPLEMENTASI

A. Perancangan Program

Program yang akan dibuat adalah sebuah program enkripsi gambar sederhana yang dikembangkan dengan bahasa pemrograman Python. Tujuan utama dari program adalah untuk memungkinkan pengguna mengenkripsi gambar sebelum diunggah ke suatu layanan *cloud storage*, sehingga meningkatkan keamanan data pribadi karena pihak penyedia layanan tidak dapat mendekripsi file gambar yang diunggah oleh pengguna. Fungsi enkripsi dan dekripsi dari program ini memanfaatkan implementasi dari fungsi chaos *Arnold's Cat Map*.

Untuk menggunakan program, pengguna akan memasukkan sebuah kunci rahasia yang berbentuk rangkaian angka ke dalam program. Kunci rahasia ini berfungsi sebagai parameter b , c , dan jumlah iterasi yang digunakan untuk proses enkripsi. Kunci yang sama juga digunakan dalam proses dekripsi gambar. Kunci rahasia tersebut disimpan oleh pengguna dalam bentuk sebuah *string*. Berikut adalah implementasi proses pengambilan parameter fungsi ACM dari kunci rahasia.

```
def parse_key(key):
    iterations = int(key[:2])
    b = int(key[2:4])
    c = int(key[4:6])
    return iterations, b, c
```

Ukuran dimensi gambar asli yang ingin dienkripsi dalam program ini tidak dibatasi oleh batasan dimensi persegi $N \times N$ dari fungsi Arnold's Cat Map. Hal ini dicapai dengan proses

pemberian *padding* pada gambar asli terlebih dahulu sebelum dilakukan enkripsi. Proses ini memungkinkan pengguna untuk mengenkripsi gambar dengan ukuran dimensi yang bebas, memberikan fleksibilitas yang lebih besar dalam penggunaan program. Berikut adalah implementasi proses *padding* gambar sebelum dienkripsi.

```
def pad_image(array):
        h, w, _ = image_array.shape
        max = max(h, w)
        padded = np.zeros((max,max,4), dtype = array.dtype)
        padded[:, :, 3] = 127
        if h > w:
            padded[:h, :w, :] = image
        else:
            padded[:h, :w, :] = image
    return padded
```

Pertama-tama gambar asli dirubah menjadi sebuah *numpy array* yang kemudian ditentukan dimensi terpanjangnya. *Array* dari gambar tersebut lalu diberikan padding pada salah satu sumbunya agar dimensi gambar menjadi persegi. Padding yang diberikan memiliki tingkat transparansi yang berbeda dengan bagian gambar aslinya untuk memudahkan proses *cropping* setelah tahap dekripsi.

Selanjutnya dilakukan proses enkripsi yang mengubah gambar asli dengan mengacak posisi setiap titik pada gambar sehingga menjadi gambar yang tidak dapat dikenali. Proses enkripsi ini melindungi informasi yang terkandung di dalam gambar asli dari akses yang tidak diinginkan oleh pengguna. Berikut adalah implementasi dari proses enkripsi dengan fungsi ACM.

```
def arnolds_cat_map(image, iterations, b, c):
    n = image.shape[0]
    transformed = image.copy()
    for i in range(iterations):
        new_image = np.zeros_like(transformed)
        for x in range(n):
            for y in range(n):
                new_x = (x + b * y) % n
                new_y = (c * x + (b * c + 1) * y) % n
                new_image[new_x, new_y] = transformed [x, y]
            transformed = new_image
    return transformed
```

Setiap titik dari gambar dipindahkan ke posisi koordinat yang baru berdasarkan fungsi ACM. Proses ini kemudian diulangi sebanyak jumlah iterasi yang didapat dari kunci rahasia. Parameter b , c pada matriks juga diambil dari kunci rahasia. Proses dekripsi memiliki alur implementasi yang mirip dengan proses enkripsi. Perbedaan utamanya terletak pada matriks yang digunakan dalam fungsi ACM. Berikut adalah implementasinya untuk fitur dekripsi gambar.

```
def arnolds_cat_map(image, iterations, b, c):
    n = image.shape[0]
    transformed = image.copy()
    for i in range(iterations):
        new_image = np.zeros_like(transformed)
        for x in range(n):
            for y in range(n):
                new_x = ((b * c + 1) * x - b * y) % n
                new_y = (-c * x + y) % n
                new_image[new_x, new_y] = transformed [x, y]
            transformed = new_image
    return transformed
```

Setelah gambar melalu proses dekripsi, gambar yang didapatkan adalah gambar yang masih memiliki padding dan belum berukuran sama dengan gambar asli. Proses *cropping* selanjutnya dilakukan setelah dilakukan proses dekripsi untuk menghasilkan gambar asli kembali dengan ukuran dimensi aslinya juga. Penentuan bagian dari gambar yang akan dibuang adalah dengan menentukan batas-batas titip gambar yang nilai transparansinya berbeda. Berikut adalah implementasi *cropping* gambar setelah didekripsi.

```
def crop_image(padded):
        alpha = padded[:, :, 3]
        non_transparent = np.where(alpha != 127)
        min_y = np.min(non_transparent[0])
        max_y = np.max(non_transparent[0])
        min_x = np.min(non_transparent[1])
        max_x = np.max(non_transparent[1])
        return padded[min_y:max_y + 1,
                min_x:max_x + 1, :]
```

Selain bagian-bagian utama dari program tersebut, dikembangkan pula sebuah antarmuka sederhana untuk memudahkan penggunaan program. Antarmuka ini masih berbentuk menu di CLI karena program akan dijalankan secara lokal di perangkat pengguna.

```
PS F:\kripto-cat-map> python encryption.py

Enkripsi/Dekripsi Gambar menggunakan Arnold's Cat Map
Menu
1. Enkripsi gambar
2. Dekripsi gambar
3. Keluar
Masukkan nomor menu:
```

Gambar 2 Tampilan Menu Awal Program

Gambar XX menampilkan menu dari program ini. Fitur-fitur program dapat diakses dengan memasukkan kode nomor menu. Bila pengguna memilih kode 1, maka program akan meminta pengguna untuk memasukkan lokasi file gambar yang akan dienkripsi beserta dengan *string* kunci rahasia. Bila pengguna memilih kode 2, maka program akan meminta pengguna untuk kembali memasukkan lokasi file gambar terenkripsi yang ingin didekripsi berserta kunci rahasia yang digunakan dalam proses enkripsi gambar tersebut. Kode 3 digunakan untuk menutup dan keluar dari program. Apabila memasukkan selain dari ketiga kode tersebut, maka program akan memberikan peringatan kode invalid dan meminta pengguna untuk memasukkan kode lain.

IV. PEMBAHASAN

A. Uji Coba Program

Eksperimen dilakukan untuk menguji fungsi enkripsi dan dekripsi dari program terhadap gambar yang berbentuk persegi dan gambar yang berbentuk persegi panjang. Berikut adalah kedua gambar "square.png" dan "long.png" yang akan diuji coba.

Gambar 3 Gambar Asli square.png

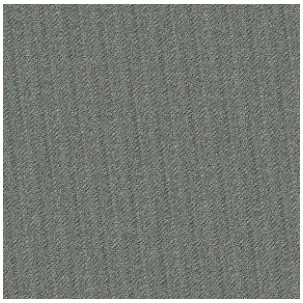
Gambar 4 Gambar Asli long.png

Pada percobaan pertama dilakukan proses enkripsi dan dekripsi secara normal terhadap kedua gambar asli terlebih dahulu. Kunci rahasia yang digunakan adalah "120405" di mana parameter b , c , dan jumlah iterasi yang akan digunakan masing-masing adalah 4, 5, dan 12.

```
Enkripsi/Dekripsi Gambar menggunakan Arnold's Cat Map
Menu
1. Enkripsi gambar
2. Dekripsi gambar
3. Keluar
Masukkan nomor menu: 1
Masukkan path menuju gambar: square.png
Masukkan kunci: 120405
Enkripsi selesai. Gambar disimpan sebagai square_encrypted.png.


Enkripsi/Dekripsi Gambar menggunakan Arnold's Cat Map
Menu
1. Enkripsi gambar
2. Dekripsi gambar
3. Keluar
Masukkan nomor menu: 1
Masukkan path menuju gambar: long.png
Masukkan kunci: 120405
Enkripsi selesai. Gambar disimpan sebagai long_encrypted.png.
```

Gambar 5 Gambar Hasil Dekripsi long_encryted.png

Hasil dari proses enkripsi kedua gambar adalah file "square_encrypted.png" dan "long_encrypted.png" sebagai berikut.



Gambar 6 Gambar Hasil Enkrispi dari square.png

Gambar 7 Gambar Hasil Enkripsi dari long.png

Selanjutnya dilakukan proses dekripsi terhadap kedua gambar di atas menggunakan kunci rahasia yang sama dengan yang digunakan pada tahap enkripsi. Hasilnya adalah kedua gambar berhasil didekripsi kembali menjadi gambar asli dengan ukuran dimensi yang sesuai juga.

Gambar 8 Gambar Hasil Dekripsi square_encrypted.png

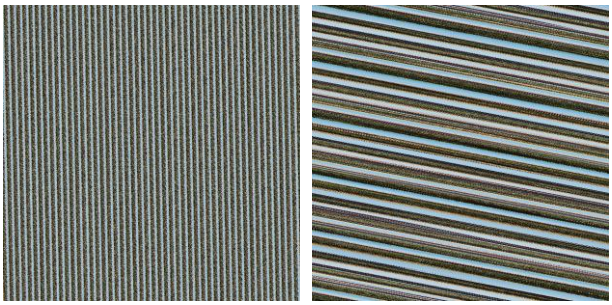
Gambar 9 Gambar Hasil Dekripsi long_encryted.png

Pada percobaan berikutnya dilakukan proses dekripsi dengan menggunakan kunci rahasia yang berbeda dengan kunci yang digunakan pada proses enkripsi gambar.

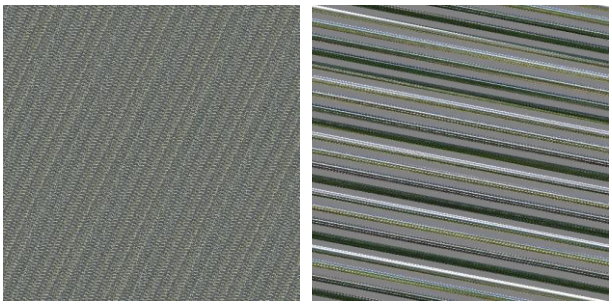
```
Enkripsi/Dekripsi Gambar menggunakan Arnold's Cat Map
Menu
1. Enkripsi gambar
2. Dekripsi gambar
3. Keluar
Masukkan nomor menu: 2
Masukkan path menuju gambar: square_encrypted.png
Masukkan kunci: 120306
Dekripsi selasai. Gambar disimpan sebagai square_encrypted_decrypted.png.

Enkripsi/Dekripsi Gambar menggunakan Arnold's Cat Map
Menu
1. Enkripsi gambar
2. Dekripsi gambar
3. Keluar
Masukkan nomor menu: 2
Masukkan path menuju gambar: long_encrypted.png
Masukkan kunci: 120306
Dekripsi selasai. Gambar disimpan sebagai long_encrypted_decrypted.png.
```

Gambar 10 Gambar Hasil Dekripsi long_encryted.png

Dengan menggunakan kunci "120306" yang mengubah parameter *b* dan *c* dari persamaan ACM, gambar yang dihasilkan dari proses dekripsi tidak sama dengan gambar asli, baik secara visual maupun dari ukuran dimensi gambar. Hal ini menunjukkan bahwa proses dekripsi gagal dilakukan. Kunci berbeda lainnya yang hanya mengubah jumlah iterasi "110405" juga digunakan sebagai perbandingan.



Gambar 11 Gambar Hasil Dekripsi square_encryted.png dengan Kunci yang Berbeda dalam Parameter b dan c (kiri) dan Iterasi (kanan)



Gambar 12 Gambar Hasil Dekripsi long_encryted.png dengan Kunci yang Berbeda dalam Parameter b dan c (kiri) dan Iterasi (kanan)

B. Analisis

Fungsionalitas program dalam melakukan enkripsi dan dekripsi file gambar berjalan sesuai dengan harapan. Proses enkripsi dan dekripsi memerlukan kunci rahasia yang sama untuk dapat berfungsi dengan baik. Penggunaan kunci rahasia yang berbeda, meskipun hanya sedikit berbeda, akan menyebabkan proses dekripsi gagal. Ini menunjukkan betapa pentingnya konsistensi dalam penggunaan kunci rahasia untuk menjaga integritas proses enkripsi dan dekripsi.

Hal ini semakin diperjelas terutama pada file gambar dengan ukuran dimensi yang bukan persegi. Dengan adanya tahapan *padding*, ukuran dimensi gambar asli tidak dapat diketahui dari gambar terenkripsinya. Padding memastikan bahwa gambar terenkripsi tidak memberikan petunjuk apa pun tentang dimensi aslinya, menambah lapisan keamanan ekstra terhadap analisis pihak ketiga yang mencoba mengungkap informasi dari gambar terenkripsi. Sehingga, program berhasil menjalankan tujuannya untuk menambahkan lapisan keamanan dan privasi pada file gambar milik pengguna.

V. KESIMPULAN DAN SARAN

A. Kesimpulan

Fungsi *chaos* Arnold's Cat Map dapat digunakan dalam implementasi fitur mengenkripsi dan mendekripsi file gambar dengan baik. Fitur ini dapat menjadi lapisan keamanan dan privasi yang baik di sisi pengguna dalam menggunakan layanan *cloud storage*. Pihak selain pengguna yang mengetahui kunci rahasia enkripsi gambar tidak dapat melakukan dekripsi pada gambar yang terenkripsi. Fungsi Arnold's Cat Map menjadi salah satu pilihan yang baik karena perbedaan kecil dalam nilai kunci rahasia tetap membuat proses dekripsi tidak dapat dilakukan. Selain itu, tidak dibutuhkan jumlah iterasi yang besar untuk mendapatkan gambar yang teracak dengan baik.

B. Saran

Implementasi program dalam makalah ini masih berfokus pada kasus pengguna tunggal yang ingin menyimpan data gambarnya di *cloud storage*, sehingga protokol pertukaran dan pembangkitan kunci rahasia tidak dilakukan. Penelitian selanjutnya dapat menyelidiki cara-cara pertukaran dan pembangkitan kunci rahasia yang aman untuk digunakan dalam kolaborasi beberapa pengguna. Penggunaan fungsi *chaos* lainnya bersamaan dengan Arnold's Cat Map juga dapat ditelusuri lebih jauh lagi.

VI. UCAPAN TERIMA KASIH

Penulis mengucapkan rasa syukur kepada Allah SWT. atas segala rahmat dan karunia-Nya yang telah memungkinkan penulis untuk menyelesaikan makalah yang berjudul "Implementasi Fitur Enkripsi Gambar dengan Arnold's Cat Map pada Layanan Cloud Storage untuk Menambah Lapisan Keamanan" ini. Penulis ingin menyampaikan terima kasih yang sebesar-besarnya kepada Bapak Rinaldi Munir, selaku dosen mata kuliah IF4020 Kriptografi, atas segala bimbingan dan ilmu yang telah diberikan. Ucapan terima kasih juga penulis sampaikan kepada orang tua dan teman-teman penulis yang

selalu memberikan dukungan dan semangat selama proses perkuliahan.

REFERENSI

[1] Munir, Rinaldi. *Pembangkit Bilangan Acak*. Program Studi Teknik Informatika. 2024.
Diakses pada 8 Juni 2024
https://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/20232024/32-Pembangkit-bilangan-acak-2024.pdf

[2] *A clarification on Adobe Terms of Use*. Adobe. 2024.
Diakses pada 10 Juni 2024
https://blog.adobe.com/en/publish/2024/06/06/clarification-adobe-terms-of-use

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 Juni 2024

Muhammad Rakha Athaya
13520108