Jurnal Masyarakat Informatika *ISSN: 2086-4930*
Vol. 14, No. 1, 2023 *e-ISSN: 2777-0648*

Image Encryption and Decryption Using Vigenere Cipher with Compute Unified Device Architecture (CUDA)

Arjuna Wahyu Kusuma[1)], R. Damanhuri[2)], Muhamad Nur Baihaqi[3)], Labib Habibie Sanjaya[4)]

Departemen Ilmu Komputer/Informatika, Fakultas Sains dan Matematika, Universitas Diponegoro
[1)]arjunawahyukusuma@students.undip.ac.id, [2)]rdamanhuri@students.undip.ac.id,
[3)]baihaqi2193@students.undip.ac.id, [4)]labibhs@students.undip.ac.id

Abstrak

Compute Unified Device Architecture (CUDA) adalah Application Programming Interface (API) NVIDIA dan platform yang memungkinkan akses langsung ke set instruksi GPU dan memberi dukungan untuk berinteraksi dengan GPU terkait komputasi paralel. Dengan CUDA, komputasi yang kompleks menjadi lebih cepat dan lebih efisien. Vigenere Cipher adalah kriptografi klasik populer yang mengimplementasikan kunci simetris dengan panjang tertentu. Pada penelitian ini, penerapan enkripsi dan dekripsi Vigenere Cipher dilakukan pada citra serta dengan CPU dan GPU (CUDA). Paralelisasi dengan CUDA menunjukkan hasil eksekusi waktu yang relatif lebih cepat daripada CPU. Persentase rata-rata penurunan waktu adalah 99,46 persen untuk enkripsi serta 99,47 persen untuk dekripsi.

Kata kunci: *CPU, CUDA, GPU, Image, Vigenere Cipher*

Abstract

Compute Unified Device Architecture (CUDA) is NVIDIA's Application Programming Interface (API) and platform that enables direct access to the GPU instruction set and provides support for interfacing with the GPU regarding parallel computing. With CUDA, complex computations become faster and more efficient. The Vigenere Cipher is a popular classic cryptography that implements a symmetric key of a certain length. In this study, the application of Vigenere Cipher encryption and decryption was carried out on images and with CPU and GPU (CUDA). Parallelization with CUDA results in a relatively faster execution time than the CPU. The average percentage of time reduction is 99.46 percent for encryption and 99.47 for decryption.

Keywords: *CPU, CUDA, GPU, Image, Vigenere Cipher*

1 PENDAHULUAN

Pada zaman modern ini, informasi banyak tersebar di mana-mana. Beberapa di antaranya dapat diakses oleh publik, sementara yang lain bersifat rahasia. Informasi yang rahasia memerlukan penanganan khusus dalam penyimpanan dan pengaksesannya. Untuk menjaga kerahasiaan suatu informasi, kriptografi dapat diterapkan.

Kriptografi adalah teknik menyamarkan data atau informasi menjadi bentuk yang tidak dimengerti untuk menjaga kerahasiaan, integritas, keaslian, dan autentikasi pesan [1]. Secara umum, proses di dalamnya terbagi menjadi dua, yaitu enkripsi (*encryption*) dan dekripsi (*decryption*). Enkripsi adalah proses menyembunyikan informasi, sedangkan dekripsi adalah proses ekstraksi informasi dari informasi yang dienkripsi.

Data atau informasi yang dapat dienkripsi dan didekripsi tidak hanya yang berbentuk teks. Beragam bentuk data atau informasi lain, seperti citra, audio, dan video juga dapat dikenai proses kriptografi.

Saat ini, citra dianggap sebagai sumber informasi yang paling penting [2]. Perlindungan informasi yang berbentuk citra pun menjadi krusial. Citra merupakan kumpulan piksel yang memiliki nilai intensitas yang berbeda. Setiap citra terdiri dari n*m jumlah piksel, dengan n adalah jumlah baris dan m adalah jumlah kolom. Piksel (elemen gambar) adalah blok kecil yang mewakili jumlah intensitas abu-abu yang akan ditampilkan untuk bagian gambar tertentu [3].

Citra dapat dilindungi dengan kriptografi menggunakan beberapa algoritma yang ada, salah satunya ialah Vigenere Cipher. Vigenere Cipher termasuk *polyalphabetic substitution* yang bekerja berdasarkan Caesar Cipher dan menggunakan Vigenere Square. Algoritma ini merupakan kriptografi klasik yang mengimplementasikan kunci simetris dengan panjang tertentu. Jika panjang kunci kurang dari panjang *plaintext*, kunci tersebut akan diulang secara periodik hingga memiliki panjang sama dengan *plaintext*.

Umumnya, Vigenere Cipher digunakan untuk kriptografi pada domain teks. Namun, beberapa penelitian telah memakai algoritma ini untuk perlindungan pada citra. Pengaplikasian proses enkripsi dan dekripsinya dilakukan pada representasi citra berupa matriks numerik.

Kriptografi pada citra lebih memakan waktu daripada teks karena ukurannya tidak sekecil *file* teks. Hal ini terjadi karena lama eksekusi kriptografi dipengaruhi ukuran objek yang dienkripsi atau didekripsi. Maricar dan Sastra [4] mengungkapkan bahwa makin besar *file* yang dienkripsi, makin lama waktu prosesnya.

Masalah ini dapat diatasi dengan penerapan pemrograman paralel menggunakan Compute Unified Device Architecture (CUDA). CUDA adalah salah satu standar antarmuka dalam pemrograman paralel yang diterapkan ke dalam Graphics Processing Unit (GPU) NVIDIA. GPU adalah prosesor dari banyak inti yang lebih kecil dan lebih terspesialisasi. Dengan implementasi CUDA, komputasi yang kompleks dengan lebih cepat dan lebih efisien ditangani daripada dengan CPU [5].

Pada kriptografi citra dengan CUDA, setiap input dibagi menjadi bagian-bagian yang lebih kecil untuk diproses bersamaan (secara paralel) melalui beberapa unit pemrosesan independen. Hal ini akan menurunkan waktu enkripsi maupun dekripsi untuk resolusi gambar dan ukuran video yang tinggi [6].

2 TINJAUAN PUSTAKA

2.1 PENELITIAN TERKAIT

Mohammad, Rahim, Zeebaree, dan Ahmed [2] telah melakukan penelitian tentang *image encryption* untuk citra menggunakan beberapa algoritma, termasuk Vigenere Cipher. Hasilnya menunjukkan bahwa Vigenere Cipher termasuk yang sangat cepat pemrosesannya dibandingkan algoritma lain (0,01634 detik). Namun, objek penelitian tersebut hanya terbatas pada citra Lena berukuran 256x256.

Riadi, Fadlil, dan Tsani [7] telah mengimplementasikan Vigenere Cipher dengan pengubahan format citra menjadi *encoding* base64 dan penggunaan *tabula recta* berisi susunan huruf radix-64 untuk proses *encoding* base64. Proses enkripsi membutuhkan waktu kurang dari 0,2 detik, sementara dekripsi 0,19 detik.

Bharadwaj, Banu, Madiajagan, Ghalib, Castillo, dan Shankar [8] melakukan implementasi *image encryption* dengan akselerasi GPU. Algoritma yang digunakan adalah modifikasi XOR Cipher. Hasilnya, rata-rata rasio *speed-up* antara pemrograman serial dan paralel GPU adalah 3,489 untuk enkripsi dan 4,055 untuk dekripsi.

Darari, Winarko, dan Damayanti [9] menggunakan Vigenere Cipher dan RSA untuk enkripsi dan dekripsi *image.* Tabel Vigenere yang digunakan telah dimodifikasi dari versi klasik menjadi berisi 256 baris dan kolom menyesuaikan intensitas piksel citra, yakni 0–255.

Penelitian lain oleh Elrefaey, Sarhan, dan El-Shennawy [6] mengimplementasikan pemrograman paralel GPU CUDA-OpenGL pada enkripsi *chaotic-maps-based* untuk citra dan video. Hasilnya, waktu eksekusi berkurang hampir 75% untuk kecepatan enkripsi.

2.2 VIGENERE CIPHER

Vigenere cipher adalah metode untuk mengenkripsi teks alfabet. Metode ini menggunakan bentuk sederhana dari subtitusi polialfabetik. Sebuah sandi polialfabetik merupakan sandi apa pun yang didasarkan pada subtitusi, menggunakan beberapa abjad subtitusi. Enkripsi teks asli dilakukan menggunakan Vigenere square atau tabel Vigenère.

Pada versi klasik, tabel yang digunakan hanya mewakili 26 abjad. Hal ini dinilai menjadi kekurangan algoritma vigenere cipher sehingga muncul versi tabel yang telah modifikasi untuk menyesuaikan domain penelitian dan meningkatkan keamanan. Salah satunya adalah tabel modifikasi dengan isi 256 baris dan kolom yang mewakili intensitas piksel citra [9]. Perbedaan dari versi klasik hanya terdapat pada jumlah baris dan kolom. Setiap karakter tetap dipindahkan secara siklis ke kiri dibandingkan dengan karakter sebelumnya, yang sesuai dengan 256 kunci yang mungkin.

Pada titik-titik yang berbeda dalam proses enkripsi, sandi menggunakan karakter yang berbeda dari salah satu baris. Karakter yang digunakan pada setiap titik tergantung pada kata kunci yang terus-menerus.

Langkah yang dilakukan adalah mengubah sebuah citra menjadi *array*. Hal ini berarti memecah citra menjadi piksel-piksel individu dan tiap piksel diwakili oleh sejumlah angka sehingga menghasilkan *array* yang sangat besar. Lalu, dilakukan enkripsi terhadap *array* tersebut menggunakan sandi Vigenere.

2.3 PARALLEL PROCESSING

Pemrosesan paralel adalah komputasi yang melakukan banyak tugas secara bersamaan. Ini memungkinkan komputer untuk melakukan banyak operasi pada saat yang sama, bukan secara berurutan, dengan membagi beban kerja di antara banyak prosesor atau inti. Ini dapat secara signifikan meningkatkan kecepatan dan efisiensi jenis perhitungan tertentu, terutama yang dapat dipecah menjadi tugas-tugas independen.

Ada beberapa cara untuk mengimplementasikan pemrosesan paralel, termasuk menggunakan banyak prosesor atau inti dalam satu komputer, menggunakan banyak komputer yang terhubung melalui jaringan, atau menggunakan perangkat keras khusus seperti unit pemrosesan grafik (GPU).

2.4 GRAPHICS PROCESSING UNIT

Graphics Processing Unit (GPU) adalah jenis prosesor khusus yang dirancang khusus untuk menangani tugas-tugas yang berhubungan dengan grafis. Ini digunakan di komputer, ponsel cerdas, dan perangkat lain untuk merender gambar dan video, serta untuk melakukan tugas lain yang memerlukan pemrosesan berkecepatan tinggi dan kemampuan pemrosesan paralel.

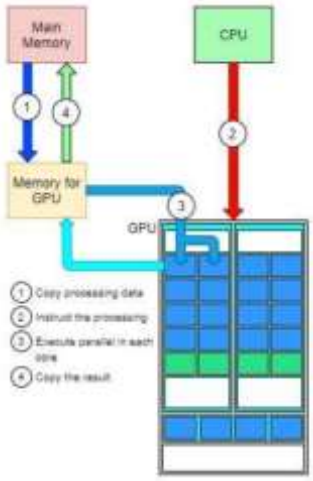
GPU dibangun menggunakan ribuan inti pemrosesan kecil dan efisien yang dapat melakukan banyak tugas secara bersamaan. Hal ini memungkinkan adanya penanganan operasi grafis yang kompleks jauh lebih cepat daripada unit pemrosesan pusat (CPU) tradisional, yang dirancang untuk menangani berbagai tugas tujuan umum.

Selain tugas grafik tradisional, GPU semakin banyak digunakan untuk melakukan berbagai tugas nongrafis yang dapat memanfaatkan kemampuan pemrosesan paralelnya.

2.5 COMPUTE UNIFIED DEVICE ARCHITECTURE

CUDA adalah platform komputasi paralel dan Application Programming Interface (API) NVIDIA yang memungkinkan akses langsung ke set instruksi GPU dan memberi dukungan untuk berinteraksi dengan GPU terkait komputasi paralel [9].

Arsitektur CUDA terdiri dari tiga bagian dasar yang membantu pengembang sistem secara efisien memanfaatkan kemampuan komputasi kartu grafis (NVIDIA) [5]. CUDA membagi device ke grid, block, dan thread dalam struktur. Satu GPU berisi sejumlah grid, satu grid berisi sejumlah block, dan satu block berisi sejumlah thread [10]. Gambar 1 menampilkan alur proses CUDA.



Gambar 1 Alur Proses CUDA [5]

3 METODE PENELITIAN

Penelitian ini dilaksanakan dengan mengenkripsi dan mendekripsi citra memakai Vigenere Cipher. Implementasi formula ini dicatat dan hasilnya akan dibandingkan antara penggunaan CPU (iterasi tiap piksel) dan GPU (menggunakan CUDA). Variabel independen untuk penelitian ini adalah berbagai ukuran resolusi gambar keluaran dan dua proses kriptografi (enkripsi dan dekripsi).

Tahapan penelitian ini terdiri dari desain algoritma, implementasi algoritma, eksperimen, dan evaluasi.

3.1 DESAIN ALGORITMA

Pada algoritma Vigenere Cipher, terdapat kunci (key) yang digunakan untuk melakukan proses deskripsi serta enkripsi. Dalam proses enkripsi dan dekripsi, perlu diterapkan juga kunci yang sama agar hasil input dan output sama. Dalam Vigenere Cipher, kunci tersebut disebut dengan Vigenere Tableau. Rumus enkripsi dan dekripsi dari Vigenere Cipher berbeda. Berikut adalah rumus enkripsi dari Vigenere Cipher:

$$E_{ki}(a) = (a + k_i) mod\ 256 \quad (1)$$

Keterangan:
$E_{ki}(a)$: *Cipher image* pixel untuk indeks ke-a
a : Indeks pixel plain image
k_i : Penambahan posisi data dalam urutan nlai kunci

Pada proses dekripsi, digunakan rumus sebagai berikut:

$$E_{ki}(a) = (a - k_i) mod\ 256 \quad 2)$$

Keterangan:

$E_{ki}(a)$: *Cipher image* pixel untuk indeks ke-a
a : Indeks pixel plain image
k_i : Pengurangan posisi data dalam urutan nlai kunci

Sebelum melakukan enkripsi atau dekripsi, setiap warna piksel dalam gambar dipisahkan menjadi 3 saluran *(channel warna)* yaitu R *(red),* G *(green),* dan B *(blue)*. Warna piksel tersebut dipisahkan dan diiterasikan untuk dilakukan enkripsi dengan menggunakan kunci yang sama. Algoritma enkripsi yang digunakan dalam penelitian ini adalah sebagai berikut:

```
Input: image_input[]
Output:  image_output[]

Key = [50,100]
N = length(key)
For every x until(image_input[0].size)do
For every y until(image_input[1].size)do
For every z until(image_input[2].size)do
an_array[x][y][z]=(an_array[x][y][z]  + key[((x+1)*(y+1))% n])% 256
```

3.2 IMPLEMENTASI ALGORITMA

Untuk setiap algoritma enkripsi dan dekripsi, dipilih dua buah key yang bernilai 50 dan 100 sehingga key tersebut memiliki panjang 2. Untuk setiap algoritma, dibuat juga kode untuk masing-masing runtime CPU dan GPU. Pada algoritma CPU, iterasi dilakukan dengan menggunakan perulangan *for* untuk setiap dimensi *channel* dari image hingga akhir dari ukuran channel tersebut. Hal tersebut berbeda dengan algoritma GPU yang dalam implementasinya, menggunakan fungsi grid untuk mengambil pixel sesuai dengan indeks processing unit dari GPU hingga akhir dari ukuran image dalam suatu channel warna.

Dalam implementasinya, kami menggunakan Google Colab untuk menjalankan runtime GPU serta CPU. Bahasa yang digunakan untuk mengimplementasikan algoritma ini adalah bahasa Python. Digunakan juga libary numba untuk melakukan komputasi yang memanfaatkan GPU. Berikut adalah spesifikasi CPU dan GPU yang digunakan pada runtime Google Colab:

Tabel 1 Spesifikasi CPU

Nama CPU	Jumlah *Core*	Jumlah *Thread*	*Clock Speed Rate* (MHz)	Memori Sistem (RAM)
Intel(R) Xeon(R) CPU @ 2.30GHz	2	1	2300 Mhz	13 GB

Tabel 2 Spesifikasi GPU

Nama GPU	Jumlah CUDA *Core*	*Clock Speed Rate* (MHz)	Memori Sistem (RAM)
NVIDIA Tesla T4 GPU	2560	1590 Mhz	16 GB

3.3 EKSPERIMEN

Pada penelitian ini terdapat beberapa skenario uji yang digunakan sebagai eksperimen terhadap program. Pembeda setiap skenario adalah ukuran citra yang diproses dan proses kriptografi yang dilakukan. Ukuran citra yang digunakan adalah citra 256 x 256, 512 x 512,

1024 x 1024, dan 2048 x 2048. Sementara itu, proses kriptografi yang diujikan adalah enkripsi dan dekripsi. Untuk menghindari bias waktu tambahan inisiasi kernel pada GPU (CUDA) saat pertama kali dijalankan, dilakukan pemrosesan terhadap citra 128 x 128 tambahan sebelum pengujian skenario utama. Citra tambahan ini tidak dimasukkan dalam perhitungan dan analisis skenario uji.

3.4 EVALUASI

Keluaran yang akan dianalisis adalah total waktu eksekusi yang dibutuhkan oleh CPU dan GPU (CUDA). Setelah mendapatkan keluaran untuk setiap skenario tes, data hasil akan dibandingkan dan dibuat *line chart* sebagai representasi statistik dengan sumbu x adalah ukuran citra dan sumbu y adalah waktu eksekusi. *Chart* dalam penskalaan logaritmik.

4 HASIL DAN PEMBAHASAN

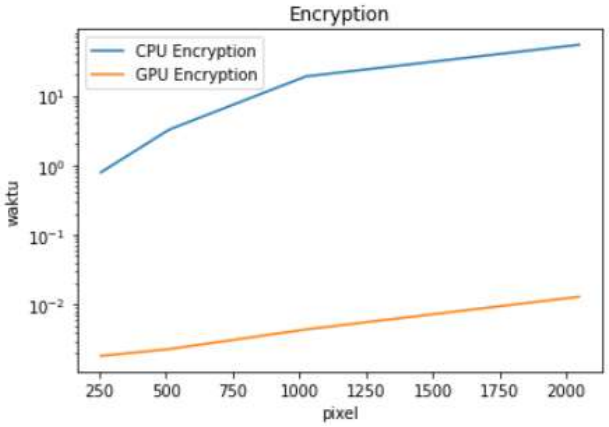
4.1 ENKRIPSI

Untuk seluruh variabel ukuran citra, eksekusi enkripsi menggunakan GPU lebih cepat daripada CPU. Persentase rata-rata penurunan waktu untuk 4 skenario uji adalah 99,46 persen.

Untuk CPU dan GPU, makin besar ukuran citra, makin lama waktu eksekusi yang dibutuhkan. Kenaikan waktu eksekusi antara satu ukuran citra dengan ukuran citra lain untuk CPU rata-rata 4,24 kali lipat, sedangkan untuk GPU 2,03 kali lipat.

Tabel 3 Perbandingan Waktu Enkripsi CPU dan GPU

Ukuran Citra	CPU (ms)	GPU (ms)
256 x 256	0.7908763885498047	0.001846313476562500
512 x 512	3.2494440078735350	0.002312183380126953
1024 x 1024	18.791830301284790	0.004419565200805664
2048 x 2048	53.499336242675780	0.013016462326049805



Gambar 2 Grafik Perbandingan Waktu Enkripsi CPU dan GPU

4.2 DEKRIPSI

Untuk seluruh variabel ukuran citra, eksekusi enkripsi menggunakan GPU lebih cepat daripada CPU. Persentase rata-rata penurunan waktu untuk 4 skenario uji adalah 99,47 persen.

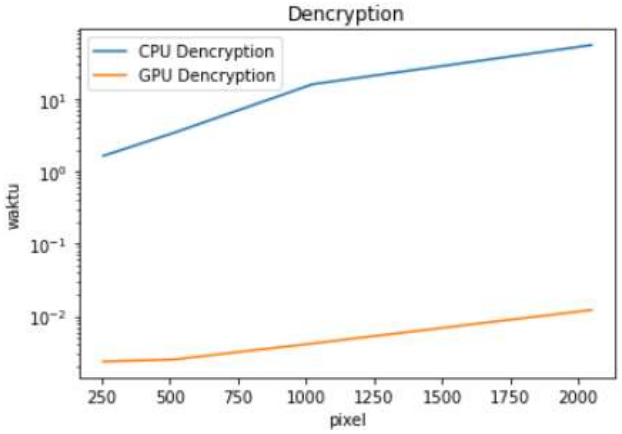
Untuk CPU dan GPU, makin besar ukuran citra, makin lama waktu eksekusi yang dibutuhkan. Kenaikan waktu eksekusi antara satu ukuran citra dengan ukuran citra lain untuk CPU rata-rata 3,42 kali lipat, sedangkan untuk GPU 1,87 kali lipat.

Selain itu, terlihat juga bahwa proses enkripsi pada CPU lebih cepat daripada proses dekripsinya untuk 3 skenario uji ukuran citra. Adapun pada GPU, proses enkripsi lebih cepat daripada dekripsinya untuk 2 skenario uji ukuran citra.

Secara keseluruhan, CPU membutuhkan waktu makin lama untuk menerapkan kriptografi Vigenere Cipher pada citra jika tugas makin banyak, sedangkan GPU (CUDA) perlu waktu eksekusi yang relatif lebih stabil.

Tabel 4 Perbandingan Waktu Dekripsi CPU dan GPU

Ukuran Citra	CPU (ms)	GPU (ms)
256 x 256	1,6382229328155518	0,0023529529571533203
512 x 512	3,3716492652893066	0,0025088787078857420
1024 x 1024	15,966381311416626	0,0041801929473876950
2048 x 2048	55,555071830749510	0,0121378898620605470



Gambar 3 Grafik Perbandingan Waktu Dekripsi CPU dan GPU

5 KESIMPULAN

Dari serangkaian eksperimen yang telah dilakukan dengan CPU dan GPU (CUDA) untuk kedua proses kriptografi pada tiap ukuran citra uji, disimpulkan bahwa waktu eksekusi enkripsi dan dekripsi Vigenere Cipher lebih cepat dengan penggunaan CUDA daripada CPU. Untuk enkripsi, persentase rata-rata penurunan waktu adalah 99,46 persen, sedangkan untuk dekripsi 99,47 persen. Hal ini terjadi karena pengolahan setiap piksel saat penyandian dilakukan secara paralel. Tidak terjadi anomali yang menunjukkan bahwa CPU mengeksekusi lebih cepat daripada GPU (CUDA) untuk citra yang diujikan.

Namun, kemungkinan CPU lebih cepat daripada GPU (CUDA) tetap ada untuk ukuran citra yang tidak terlalu besar mengingat GPU (CUDA) perlu waktu untuk mengatur paralelisasi jumlah tugas dahulu, sedangkan CPU tidak memerlukan hal tersebut.

DAFTAR PUSTAKA

[1] A. Fadlil, I. Riadi and A. Nugrahantoro, "Kombinasi Sinkronisasi Jaringan Syaraf Tiruan dan Vigenere Cipher untuk Optimasi Keamanan Informasi," *Jurnal Teknologi Informasi & Komunikasi Digital Zone,* vol. 11, no. 1, pp. 81-95, Mei 2020.

[2] O. F. Mohammad, M. S. M. Rahim, S. R. M. Zeebaree and F. Y. Ahmed, "A Survey and Analysis of the Image Encryption Methods," *International Journal of Applied Engineering Research,* vol. 12, no. 23, pp. 14265-13280, 2017.

[3] A. Susanto, T. Khotimah, M. T. Sumadi, J. Warsito and Rihartanto, "Image encryption using vigenere cipher with bit circular shift," *International Journal of Engineering & Technology,* pp. 62-64, 2018.

[4] M. A. Maricar and N. P. Sastra, "Efektivitas Pesan Teks dengan Cipher Substitusi, Vigenere Cipher, dan Cipher Transposisi," *Majalah Ilmiah Teknologi Elektro,* vol. 17, no. 1, 2018.

[5] V. M. Siregar, K. Anaga, T. M. Arkan, D. P. Pakpahan, R. Ariefftah, A. Sugiharto and F. A. Nugroho, "Efficient Computation of Mandelbrot Set Generation with Compute Unified Device Architecture (CUDA)," *2022 6th International Conference on Informatics and Computational Sciences (ICICoS),* pp. 1-5, 2022.

[6] A. Elrefaey, A. Sarhan and N. M. El-Shennawy, "Parallel approaches to improve the speed of chaotic-maps-based," *Journal of Real-Time Image Processing,* pp. 1897-1906, 2021.

[7] I. Riadi, A. Fadlil and F. A. Tsani, "Pengamanan Citra Digital Berbasis Kriptografi Menggunakan," *JISKA (Jurnal Informatika Sunan Kalijaga),* vol. 7, no. 1, pp. 33-45, Januari 2022.

[8] B. Bharadwaj, J. S. Banu, M. Madiajagan, M. R. Ghalib, O. Castillo and A. Shankar, "GPU-Accelerated implementation of a genetically optimized image," *Soft Computing,* pp. 14413-14428, 2021.

[9] R. Darari, E. Winarko and A. Damayanti, "Encryption and Decryption Application on Images with Encryption and Decryption Application on Images with," *Contemporary Mathematics and Applications,* vol. 2, no. 1, pp. 109-117, 2020.

[10] M. Cal`ı and V. D. Mauro, "Performance Analysis of Roberts Edge Detection Using CUDA and OpenGL," in *CEUR Workshop Proceedings*, 2016.

[11] B. Kurniawan, N. A. Setiawan and T. B. Adji, "Analisis Perbandingan Komputasi GPU dengan CUDA dan Komputasi CPU untuk Image dan Video Processing," *Seminar Nasional Aplikasi Teknologi Informasi (SNATi),* 2015.