

## UJI PERFORMA KOMPUTASI PARALEL ALGORITMA DISCRETE COSINE TRANSFORM UNTUK KOMPRESI CITRA

Megastin M. Lumembang<sup>1)</sup>, Edwin Tenda<sup>2)</sup>, Eliasta Ketaren<sup>3)</sup>, Andrew Lengkong<sup>4)</sup>

<sup>1,2,3)</sup>Program Studi Sistem Informasi Universitas Sam Ratulangi; Jl.Kampus, Kec. Malalayang, Kota Manado

<sup>4)</sup>Program Studi Informatika, Universitas Prisma, Jl. Pummorow, Tikala, Kota Manado

email: megastinml@unsrat.ac.id<sup>1)</sup>, tenda.edwin@unsrat.ac.id<sup>2)</sup>, eliasketaren@unsrat.ac.id<sup>3)</sup>, andreuw.lengkong@prisma.ac.id<sup>4)</sup>

### Abstrak

Kompresi data adalah cara untuk mengurangi ukuran dari berkas yang akan ditransmisikan maupun disimpan. Secara umum terdapat dua jenis kompresi data yaitu lossy compression dan lossless compression. Algoritma Discrete Cosine Transform (DCT) adalah salah satu teknik lossy compression yang banyak di gunakan dalam kompresi citra. Komputasi paralel adalah teknik komputasi yang bertujuan untuk memaksimalkan kemampuan perangkat komputasi yang memiliki lebih dari satu unit pemrosesan (CPU). Penelitian ini mengimplementasikan algoritma DCT kedalam bentuk komputasi paralel menggunakan modul multiprocessing dari bahasa Python. Hasil dari implementasi algoritma DCT kemudian di ujikan pada empat kelompok dataset dengan jumlah masing-masing sebanyak 5, 10, 15 dan 20 citra. Hasil pengujian menunjukkan bahwa implementasi algoritma DCT paralel memiliki rata-rata waktu pemrosesan 31.74% lebih baik dibandingkan dengan implementasi DCT sekuensial.

**Kata Kunci:** kompresi citra, Discrete Cosine Transform, komputasi paralel, python multiprocessing

### 1. Pendahuluan

Kompresi data adalah teknik yang digunakan untuk mengurangi ukuran berkas (file) atau data yang akan disimpan atau ditransfer. Kompresi data mengubah data masukan (input data stream, source stream, original raw data) menjadi data lain (output stream, compressed stream) yang lebih kecil ukurannya. Kompresi data membantu mengurangi waktu yang dibutuhkan mentransfer data, menghemat bandwidth dalam jaringan serta mengurangi biaya yang dibutuhkan untuk penyimpanan data. Kompresi data dapat diterapkan pada berbagai jenis berkas, baik berkas berbasis teks, gambar, audio ataupun jenis berkas lainnya. Di dalam kompresi data, faktor penting yang perlu di perhatikan adalah rasio kompresi data, jaminan kualitas data saat dilakukan dekompresi dan kecepatan dalam proses kompresi.

Pada kompresi citra, tujuan yang ingin dicapai adalah mengurangi ukuran dari citra tanpa menurunkan kualitasnya secara signifikan. Secara umum terdapat dua teknik kompresi citra, yaitu kompresi perbandingan tinggi atau Lossless compression dan kompresi perbandingan rendah atau Lossy compression. Pada lossless compression data asli dapat dikembalikan pada data semula sehingga hasil rasio dari kompresi tidak terlalu besar untuk memastikan bahwa berkas dapat dikembalikan ke keadaan semula. Metode lossless compression banyak digunakan untuk pengarsipan, dan penyuntingan data. Lossy compression merupakan teknik untuk kompresi dan dekompresi data. Metode Lossy compression menghasilkan berkas hasil kompresi yang lebih kecil dibandingkan dengan metode Lossless compression karena rasio kompresi yang lebih tinggi.

Algoritma Discrete Cosine Transform (DCT) adalah salah satu algoritma yang dapat digunakan untuk melakukan kompresi sinyal atau gambar khususnya gambar bertipe JPEG. Pada algoritma ini digunakan fungsi kosinus dalam perhitungan. Algoritma ini bekerja dengan cara melakukan konversi dari data berbentuk spasial menjadi bentuk frekuensi yang kemudian diolah dan dikonversi ulang menjadi data spasial. Algoritma ini termasuk dalam teknik kompresi Lossy compression.

Beberapa penelitian terkait algoritma DCT antara lain, Raharja et al [1] melakukan implementasi algoritme DCT untuk melakukan kompresi data pada citra berformat JPEG. Pengujian dilakukan pada 10 buah gambar dengan ukuran berbeda-beda, secara bergantian, dan kemudian mengukur ukuran dari berkas hasil kompresi. Dari pengujian diperoleh rasio kompresi rata-rata sebesar 74 %. Hamzah et al [2] mengembangkan kompresi berbasis algoritma DCT terhadap tiga format citra yaitu Joint photographic expert groups (JPEG atau JPG), Portable network graphic (PNG) dan Bitmap (BMP). Pengujian dilakukan terhadap dua buah citra yang sudah dimanipulasi kedalam tiga format tersebut. Hasil pengujian menunjukkan rata-rata rasio kompresi sebesar 50%. Sutrisman et al [3], melakukan implementasi DCT pada citra dalam Augmented Reality (AR) untuk mengetahui perbedaan kualitas citra asli dan terkompresi pada Marker based-tracking pada aplikasi Vforia dan ARCore. Pengujian menghasilkan kompresi citra sebesar 26% serta peningkatan tingkat quality score sebesar 5-40 poin. Samsuryadi [4] melakukan pengujian perbandingan metode kompresi digital DCT dan Lempel Ziv Welch (LZW). Dengan menggunakan 30 data citra keabuan dengan tiga kategori ukuran (256x256, 512x512, 1024x1024) berformat BMP

didapatkan bahwa algoritma DCT memiliki performa yang lebih baik yaitu rasio kompresi 65%. Shaheen et al [5] menerapkan algoritma DCT dan Discrete wavelet transform (DWT) untuk enkripsi pada data citra digital pada Wireless sensor network (WSN).

Komputasi sekuensial adalah suatu teknik pemrosesan data yang dilakukan dengan mengikuti urutan instruksi yang telah ditentukan. Dalam komputasi sekuensial, satu tahap proses harus selesai sebelum tahap proses berikutnya dimulai. Hal ini membuat proses pemrosesan data menjadi lebih lambat dan kurang efisien jika data yang diolah cukup besar. Sebaliknya, komputasi paralel adalah metode pemrosesan data yang dilakukan dengan mengolah data secara bersamaan atau serentak. Dalam komputasi paralel, beberapa tahap proses dapat dilakukan secara bersamaan sehingga proses pemrosesan data menjadi lebih cepat dan efisien [6]. Komputasi paralel juga dapat digunakan dalam berbagai bidang, seperti pemodelan kimia, simulasi fisika, analisis data genom, dan aplikasi pembelajaran mesin.

Beberapa penelitian terkait komputasi paralel antara lain, Marwa [7] mengembangkan enkripsi berkas PDF menggunakan algoritma MD5 dan RC4 secara paralel memanfaatkan teknologi GPU dan CUDA untuk akselerator prosesnya. Wijaya et al [8] melakukan studi kasus komputasi paralel pada penyelesaian permasalahan penghitungan matriks besar seperti menghitung le, invers dan penyelesaian persamaan linear simulas serta eigen analisis. Ma'rufi et al [9] melakukan implementasi Komputasi Paralel GPU pada Algoritme Cellular Automata (CA) menggunakan algoritma Flood fill pada teknologi GPU CUDA. Hasil pengujian menunjukkan bahwa CA GPU memiliki performa 227 kali lebih baik dibandingkan dengan CA pada CPU. Giriantari et al [10] menerapkan komputasi paralel pada aplikasi sistem informasi rumah sakit. Hasil penelitian menunjukkan bahwa proporsi komputasi unit pemrosesan terhadap data yang ada tidak proporsional sehingga terjadi keterbatasan dalam pemrosesan. Irmawati et al [11] melakukan analisis sistem komputasi paralel pada infrastruktur grid computing. Penelitian ini menggunakan beberapa komputer yang dibagi menjadi dua kluster yang terpisah. Menggunakan pustaka OpenMPI untuk lingkungan komputasi paralel dan Sun Grid Scheduler sebagai gridengine, Globus Toolkit sebagai middleware, Gridsphere dan Vine Toolkit sebagai portal grid.

Dalam penelitian ini dilakukan uji performa dari komputasi paralel algoritma DCT yang diimplementasikan menggunakan modul multiprocessing dari Bahasa pemrograman Python. Performa waktu pemrosesan algoritma DCT paralel kemudian dibandingkan dengan dengan algoritma DCT sekuensial untuk melihat kinerja dari algoritma tersebut.

## 2. Landasan Teori

### Algoritma Discrete Cosine Transform (DCT)

DCT adalah suatu metode matematika yang termasuk dalam Fast Fourier Transform (FFT). Operasi dalam proses transformasi ini adalah mengambil masukan berupa sinyal serta mentransformasikan. operasi dasar yang ditampilkan dalam transformasi ini adalah mengambil suatu signal dan mentransformasikannya dari representasi satu tipe ke tipe yang lain. transformasi ini dikerjakan secara frekuentatif dengan mengambil suatu nilai dari domain spasial dan kemudian mentransformasikannya ke dalam suatu representasi yang identik. DCT akan mengubah detail warna dari gambar asli, namun tidak begitu terlihat. pada operasi DCT digunakan nilai real [12].

Untuk menghitung DCT untuk kompresi citra digunakan persamaan DCT 2 dimensi sebagaimana pada Persamaan 1 [x].

$$F_{x,y} = \frac{C(x)C(y)}{4} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} f_{i,j} \cos \left[ \frac{(2i+1)x\pi}{2N} \right] \cos \left[ \frac{(2j+1)y\pi}{2M} \right] \quad (2.1)$$

dimana

$F_{x,y}$  = nilai DCT pada indeks (x, y)

$N, M$  = ukuran matriks

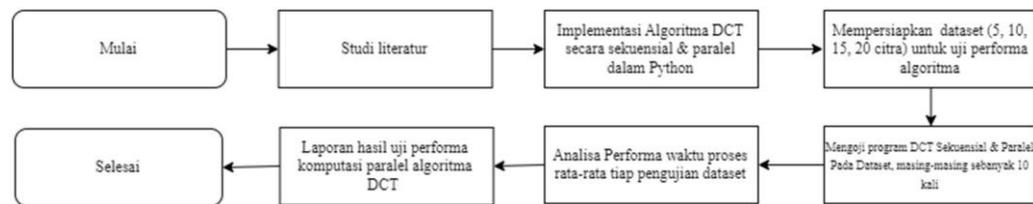
$F_{i,j}$  = nilai pixel pada index (i, j)

$$C(n) = \begin{cases} \frac{1}{\sqrt{2}}, & n = 0 \\ 1, & n \neq 0 \end{cases} ; \quad n = x, y$$

DCT bekerja dengan cara membagi gambar menjadi bagian-bagian (blok) N x N piksel. Selanjutnya Gambar dalam format RGB di konversi menjadi format YCbCr. Selanjutnya dilakukan transformasi dari blok tersebut kedalam domain frekuensi menggunakan persamaan DCT 2 Dimensi, sehingga di hasilkan representasi warna YCbCr dalam bentuk domain frekuensi. Hasil tersebut kemudian diubah menjadi fungsi matriks Cosinus yang selanjutnya diproses untuk memberihkan koefisien DCT yang tidak penting. Hasil tersebut kemudian di invers sehingga dapat dihasilkan citra yang baru.

## 3. Metode Penelitian

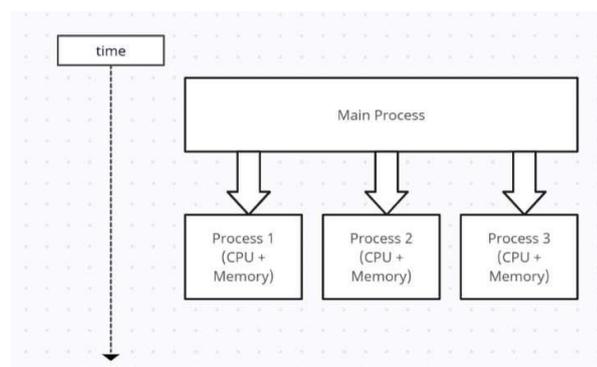
Tahapan yang dilakukan dalam penelitian ini dapat dilihat pada bagan alur pada Gambar 1



**Gambar 1.** Tahapan Penelitian

Tahapan ini dimulai dengan melakukan studi literatur terhadap algoritma DCT dan teknik komputasi paralel. Selanjutnya algoritma DCT dikembangkan menggunakan Bahasa pemrograman Python.

Pada penelitian ini komputasi paralel diimplementasikan dengan menggunakan modul Multiprocessing dalam Python. Modul Multiprocessing Python menganut Paralelisme berbasis proses [13]. Modul Multiprocessing menyediakan fungsi dan metode yang memungkinkan kita membuat dan mengelola proses secara paralel. Sehingga beberapa proses dapat dijalankan secara bersamaan. Gambar 2 menunjukkan bagaimana modul ini bekerja.



**Gambar 2.** Blok Proses Multiprocessing Python

Untuk mengukur kinerja dari algoritma DCT yang diimplementasikan secara sekuensial dan paralel dalam Bahasa pemrograman Python maka disiapkan dataset berupa kumpulan gambar dalam direktori terpisah yaitu sebanyak empat dataset dengan jumlah citra 5, 10, 15 dan 20 buah, dengan ukuran yang < 1 Mb. Selanjutnya program DCT Python sekuensial dan paralel diujikan pada masing-masing dataset sebanyak 10 kali pengulangan. Waktu proses tiap pengujian diukur untuk dilakukan pengujian. Hasil yang didapatkan selanjutnya di olah menjadi laporan hasil uji performa komputasi paralel dari algoritma DCT.

#### 4. Hasil

Program kompresi DCT yang diimplementasikan, melakukan kompresi data dengan pengaturan kualitas kompresi sedang sehingga kualitas gambar tetap terjaga namun ukuran file menjadi lebih kecil. Contoh Hasil kompresi citra menggunakan metode DCT dapat dilihat pada Gambar 3. Analisa hasil yang disajikan merupakan hasil dari waktu proses dari kompresi yang dibutuhkan dari program DCT Sekuensial dan DCT Paralel. Hasil dari pengujian pada Dataset 1 ditampilkan pada Tabel 1. Hasil dari pengujian pada Dataset 2 disajikan pada Tabel 2. Hasil dari pengujian pada Dataset 3 disajikan pada Tabel 3. Sedangkan hasil dari pengujian pada Dataset 4 disajikan dalam Tabel 4.



**Gambar 3.** Hasil Kompresi Citra menggunakan Algoritma DCT (kiri gambar asli, kanan setelah kompresi)

**Tabel 1.** Hasil Pengujian Pada Dataset 1(5 citra)

| Pengujian - ke | Waktu Proses (Detik) |             |
|----------------|----------------------|-------------|
|                | Sekuensial DCT       | Paralel DCT |
| 1              | 0,505746126          | 0,432487249 |
| 2              | 0,478854179          | 0,396760225 |
| 3              | 0,483079433          | 0,394427061 |
| 4              | 0,46213913           | 0,415902853 |
| 5              | 0,478591681          | 0,423684597 |
| 6              | 0,463803053          | 0,402308464 |
| 7              | 0,500801802          | 0,414916039 |
| 8              | 0,479351282          | 0,384077549 |
| 9              | 0,523034334          | 0,344194651 |
| 10             | 0,469816446          | 0,273360014 |
| Rata-rata      | 0,484521747          | 0,38821187  |

**Tabel 2.** Hasil Pengujian Pada Dataset 2 (10 citra)

| Pengujian - ke | Waktu Proses (Detik) |             |
|----------------|----------------------|-------------|
|                | Sekuensial DCT       | Paralel DCT |
| 1              | 2,76549745           | 1,731697798 |
| 2              | 2,80207181           | 1,811177731 |
| 3              | 2,722289085          | 1,601313829 |
| 4              | 2,763983965          | 1,615644932 |
| 5              | 2,528431177          | 1,57529664  |
| 6              | 2,35155654           | 1,539012432 |
| 7              | 2,574679852          | 1,564913034 |
| 8              | 2,405612469          | 1,560631037 |
| 9              | 2,546610355          | 1,793012857 |
| 10             | 2,549104929          | 1,611129522 |
| Rata-rata      | 2,60098376           | 1,64038298  |

**Tabel 3.** Hasil Pengujian Pada Dataset 3 (15 citra)

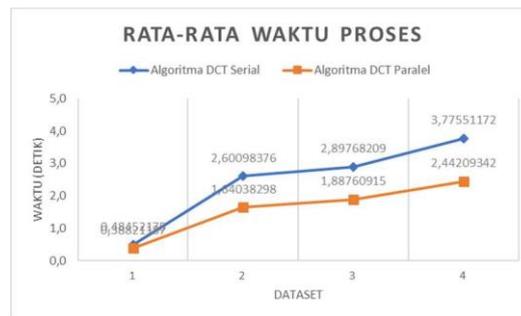
| Pengujian - ke | Waktu Proses (Detik) |              |
|----------------|----------------------|--------------|
|                | Sekuensial DCT       | Paralel DCT  |
| 1              | 2,722976446          | 2,0119881630 |
| 2              | 2,98926568           | 1,753490686  |
| 3              | 2,72322154           | 1,781115294  |
| 4              | 3,077042103          | 1,78785634   |
| 5              | 2,711264849          | 1,854910851  |
| 6              | 2,736768484          | 1,877175808  |
| 7              | 3,310594797          | 2,281341553  |
| 8              | 3,03886795           | 1,892843485  |
| 9              | 2,922519207          | 1,813382864  |
| 10             | 2,7442998886         | 1,821986437  |
| Rata-rata      | 2,897682095          | 1,887609148  |

**Tabel 4.** Hasil Pengujian Pada Dataset 4 (20 citra)

| Pengujian - ke | Waktu Proses (Detik) |             |
|----------------|----------------------|-------------|
|                | Sekuensial DCT       | Paralel DCT |
| 1              | 4,450338602          | 2,283723116 |
| 2              | 3,666678905          | 2,458423615 |
| 3              | 3,823835135          | 2,844654799 |
| 4              | 3,529760361          | 2,250577927 |
| 5              | 3,601311684          | 2,638386488 |
| 6              | 3,570009708          | 2,632920265 |
| 7              | 4,123708248          | 2,328795671 |

|           |             |             |
|-----------|-------------|-------------|
| 8         | 3,671932459 | 2,502266407 |
| 9         | 3,696532488 | 2,258545399 |
| 10        | 3,621009588 | 2,222640514 |
| Rata-rata | 3,775511718 | 2,44209342  |

Hasil dari pengujian dirangkum dan disajikan pada Gambar 4.



**Gambar 4.** Rata-rata Waktu Proses Pengujian Pada Dataset

Berdasarkan rata-rata waktu proses dari pengujian terhadap empat dataset tersebut diperoleh bahwa rata-rata waktu proses program algoritma DCT Paralel lebih cepat 31% dibandingkan dengan program algoritma DCT Sekuensial. Selain itu tampak bahwa makin besar ukuran Dataset maka makin tinggi selisih waktu proses antara program algoritma DCT Paralel dan DCT sekuensial.

## 5. Kesimpulan

Implementasi Paralel programming dapat dilakukan menggunakan modul multiprocessing pada bahasa pemrograman Python. Rata-rata waktu proses dari implementasi algoritma DCT parallel lebih baik 31% dibandingkan dengan implementasi algoritma DCT Sekuensial. Makin besar jumlah Dataset yang diuji makin tinggi selisih kinerja dari kedua implementasi algoritma DCT tersebut.

## 6. Daftar Pustaka

- [1] Bayu Dwi Raharja, 2021, Penerapan Discrete Cosine Transform (Dct) Terhadap Kompresi Citra Digital, Vol 4, No 1 (2021): Indonesian Journal of Business Intelligence (IJUBI) (31-36).
- [2] Muttaqin Md Roslan; Ahmad Fauzan bin Kadmin; Rostam Affendi Hamzah, 2021, JPG, PNG and BMP image compression using discrete cosine transform, TELKOMNIKA (Telecommunication Computing Electronics and Control), Vol 19, No 3 (hal 1010-1016).
- [3] Iwan Sutrisman; Nur Widiyasono; Heni Sulastri, 2020, Implementasi Algoritma Discrete Cosine Transform Untuk Kompresi Citra Pada Marker-Based Tracking Augmented Reality, Computatio: Journal Of Computer Science And Information Systems Vol 4, No 1 (hal 45-54).
- [4] Aprilianta, Tri Kur., Samsuryadi, Samsuryadi, 2019, Pengujian Kompresi Citra Digital Menggunakan Metode Discrete Cosine Transform (Dct) Dan Lempel Ziv Welch (Lzw). Undergraduate thesis, Sriwijaya University.
- [5] Shaheen, A.M., Sheltami, T.R., Al-Kharoubi, T.M., 2019, Digital image encryption techniques for wireless sensor networks using image transformation methods: DCT and DWT. J Ambient Intell Human Comput 10, 4733–4750.
- [6] <https://hpc.llnl.gov/documentation> (diakses pada 10 Januari 2023).
- [7] Hany, Samatra Marwa; farissi, Al, 2017, Komputasi Paralel GPU Dengan Teknologi Nvidia CUDA Untuk Enkripsi Berkas, Jurnal Sistem Informasi, Vol 9, No 2.
- [8] I Gede Pasek Suta Wijaya; Mayzar Anas; L. A. Syamsul Irfan, 2017, Studi Komputasi Paralel dan Implementasinya pada Kasus Komputasi Matriks Besar, Journal of Computer Science and Informatics Engineering (J-Cosine) Vol 1 No 1(59-64).
- [9] Ma'rufi, Muhammad Rizal; Cholissodin, Imam; Jonemaro, Eriq Muh. Adams, 2019, Implementasi Komputasi Paralel GPU pada Algoritme Cellular Automata Menggunakan CUDA®, Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer Vol 3 No 4 (3880-3887).
- [10] IA Dwi Giriantari; I Putu Adi Pradnyana Wibawa; Made Sudarma, 2018, Komputasi Paralel Menggunakan Model Message Passing Pada SIM RS (Sistem Informasi Manajemen Rumah Sakit), Jurnal Teknologi Elektro, Vol 17 No 3 (439-444).

- [11] Nelsom., Mark, 1992, The Data Compression Book, M & T Publishing Inc. Borel Avenue, San Mateo, California.
- [12] <https://docs.python.org/3/library/multiprocessing.html> (diakses pada 10 Januari 2023)