

Pembuatan Thumbnail dengan Berbagai Metode *Image Compression*

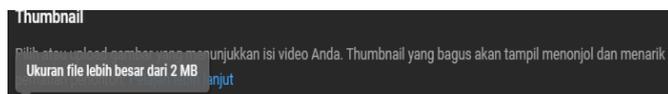
Wisnu Aditya Samiadji
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail (gmail): 13519093@std.stei.itb.ac.id

Abstrak— Penggunaan *thumbnail* atau *keluku* semakin penting seiring peningkatan penggunaan media sosial. Pembuatan *keluku* untuk media sosial tersebut biasanya disertai dengan batasan ukuran. Oleh karena itu, teknik-teknik *image compression* digunakan. Ada 2 tipe metode *image compression*, yaitu *lossy* dan juga *lossless*. Pada makalah ini, penulis akan mengimplementasikan masing-masing satu dari kedua tipe metode tersebut menggunakan MATLAB dan membandingkan tipe metode mana yang cocok untuk pembuatan *thumbnail*.

Kata Kunci—*Thumbnail, Keluku, Image Compression, Lossy, Lossless*

I. PENDAHULUAN

Teknik *image compression* semakin marak digunakan seiring meningkatnya kebutuhan media digital. Perkembangan teknologi fotografi menyebabkan ukuran citra digital yang dihasilkan semakin besar dan memakan memori yang juga besar. Sementara itu, platform media digital seringkali memiliki batasan ukuran berkas.



Buatlah ukuran gambar *thumbnail* kustom sebesar mungkin. Gambar tersebut akan digunakan sebagai gambar pratinjau dalam pemutar yang disematkan. Rekomendasi kami untuk *thumbnail* kustom antara lain:

- Memiliki resolusi sebesar 1280x720 (dengan lebar minimum 640 piksel).
- Diupload dalam format gambar seperti JPG, GIF, atau PNG.
- Tidak melebihi batas ukuran 2 MB.
- Coba gunakan rasio aspek 16:9, karena rasio ini paling sering digunakan dalam pemutar dan pratinjau YouTube.

Gambar 1. Contoh batasan pada *thumbnail* atau *keluku* di platform media digital (Youtube) (Sumber: *milik pribadi*)

Makalah ini akan memberikan gambaran aplikasi teknik *lossless image compression* dan *lossy image compression* yang akan diwakilkan oleh teknik *Huffman Coding* dan *Discrete Cosine Transform*. Implementasi kedua teknik ini akan

ditunjukkan menggunakan MATLAB yang disertai dengan GUI. Hasil implementasi kemudian akan digunakan untuk beberapa gambar yang penulis miliki untuk membandingkan ukuran gambar sebelum dan sesudah kompresi, yang kemudian akan ditentukan metode yang cocok untuk pembuatan *thumbnail*.

II. DASAR TEORI

A. Citra Digital

Citra sering disebut sebagai gambar pada bidang dwimatra (2-D). Secara matematis, citra adalah fungsi dwimatra yang menyatakan intensitas cahaya pada bidang dwimatra

$$f(x, y)$$

(x, y) : koordinat pada bidang dwimatra

$f(x, y)$: intensitas cahaya (*brightness*) pada titik (x, y)

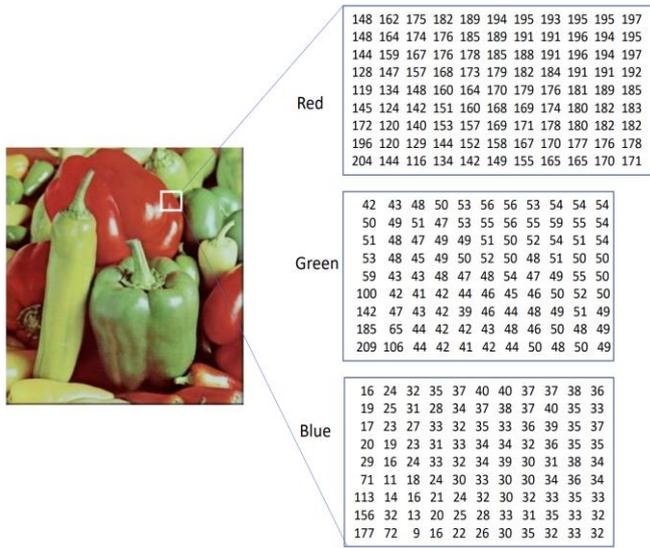
Gambar 2. Citra jika direpresentasikan secara matematis (Sumber: [01-Pengantar-Pengolahan-Citra-Bag1-2022](#))

Pada citra digital, citra direpresentasikan oleh sekumpulan pixel berbentuk grid atau matriks. Citra digital merupakan hasil pencuplikan citra kontinu secara ruang dan waktu. Jenis data yang terkandung pada citra digital adalah diskrit. Seperti yang telah disebutkan sebelumnya, citra merupakan fungsi intensitas cahaya untuk setiap x dan y pada bidang dwimatra. Hasil pencuplikan untuk citra digital dengan resolusi $M \times N$ dapat direpresentasikan seperti pada Gambar 3 berikut:

$$f(x, y) = \begin{bmatrix} f(0, 0) & f(0, 1) & \dots & f(0, N-1) \\ f(1, 0) & f(1, 1) & \dots & f(1, N-1) \\ \dots & \dots & \dots & \dots \\ f(M-1, 0) & f(M-1, 1) & \dots & f(M-1, N-1) \end{bmatrix}$$

Gambar 3. Representasi citra digital beresolusi $M \times N$ secara matematis sebagai fungsi intensitas cahaya (Sumber: [01-Pengantar-Pengolahan-Citra-Bag1-2022](#))

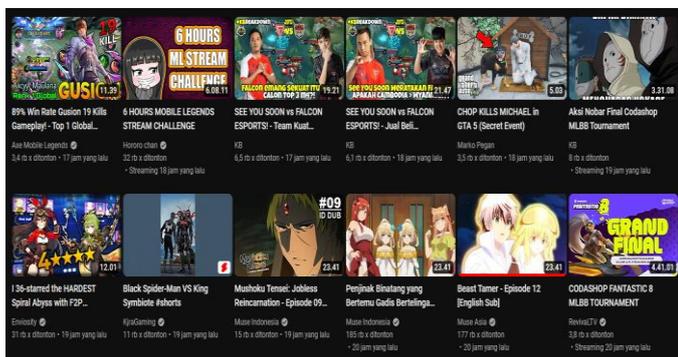
Setiap pixel pada citra digital menyimpan nilai-nilai dari derajat keabuan. Citra *grayscale* memiliki 1 nilai keabuan pada setiap pixelnya, sedangkan citra berwarna RGB memiliki 3 nilai keabuan yang masing-masing merepresentasikan derajat keabuan untuk warna merah (Red), hijau (Green), dan biru (Blue).



Gambar 4. Contoh nilai derajat keabuan untuk citra berwarna RGB (Sumber: [16-Warna-bagian1-2022](#))

B. Thumbnail

Menurut Kamus Besar Bahasa Indonesia (KBBI), *thumbnail* atau keluku adalah gambar dengan ukuran lebih kecil daripada ukuran aslinya yang ditampilkan pada monitor untuk menghemat ruang. Istilah ini biasa digunakan oleh para *graphic designer* dan fotografer. Keluku sangat berguna dalam berbagai kondisi, contohnya jika seseorang memiliki banyak sekali gambar dalam penyimpanan. Keluku dapat membantu seseorang mencari gambar tertentu dengan jauh lebih cepat pada kondisi tersebut.



Gambar 5. Thumbnail pada platform Youtube (Sumber: *milik pribadi*)

Ukuran *thumbnail* dapat bergantung pada tujuan penggunaannya. Thumbnail yang digunakan dapat berukuran sangat kecil jika tujuan utamanya hanyalah menentukan identitas gambar. Format berkas gambar yang digunakan untuk thumbnail dapat berupa JPEG, PNG, GIF, ataupun WEBP.

Format JPEG lebih umum digunakan karena ukurannya yang cenderung lebih kecil dari format berkas lainnya. Namun, resolusi dari gambar yang digunakan untuk *thumbnail* sangat bervariasi.

C. Image Compression

Image Compression atau kompresi citra merupakan operasi citra yang dilakukan dengan tujuan menghilangkan redundansi yang ditemukan pada sebuah citra. Sebuah citra biasanya perlu dikompresi karena representasi citra digital membutuhkan memori yang besar. Kompresi citra bertujuan untuk mengurangi kebutuhan memori untuk ruang penyimpanan tanpa mengurangi kualitas citra secara visual.



Gambar 6. Perbandingan citra tanpa kompresi (kiri) dan citra hasil kompresi (kanan) (Sumber : [25-Image-Compression-Bagian1-2022](#))

Ada beberapa aplikasi pemampatan citra yang digunakan hingga saat ini, diantaranya yaitu:

- Penyimpanan data di dalam media sekunder (storage) : Citra mampat membutuhkan memori di dalam storage yang lebih sedikit dibandingkan dengan citra yang tidak mampat . Contohnya adalah berkas citra berformat JPEG dibandingkan dengan citra berformat BMP.
- Pengiriman data (*data transmission*) pada saluran komunikasi data : Citra mampat membutuhkan waktu pengiriman yang lebih singkat dibandingkan dengan citra tidak mampat .Contohnya yaitu pengiriman gambar via surel, via *video conference* , via satelit luar angkasa, mengunggah gambar dari internet, dan sebagainya.

Redundansi pada sebuah citra adalah konten citra yang pada hakikatnya tidak perlu direpresentasikan dalam sejumlah bit. Ada 3 macam redundansi yang mungkin berada dalam sebuah citra:

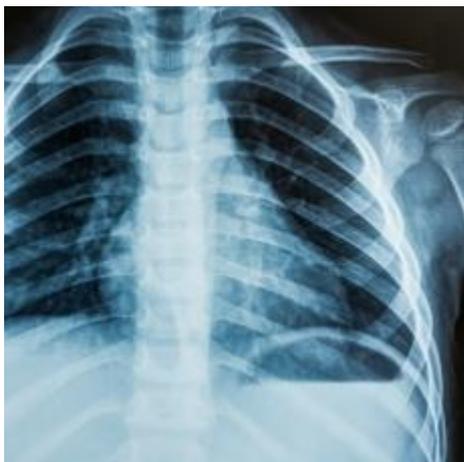
- **Coding redundancy:** Merupakan jenis redundansi yang biasanya muncul sebagai hasil pengkodean yang seragam pada setiap pixel. Tipe redundansi ini dapat dikurangi dengan mengkodekan simbol yang sering muncul dengan jumlah bit yang lebih sedikit
- **Spatial/temporal redundancy:** Merupakan jenis redundansi yang dapat muncul sebagai pixel-pixel bertetangga memiliki nilai intensitas yang tidak jauh

berbeda. Biasanya, DCT Coding digunakan untuk mengurangi redundansi tipe ini.

- **Psychovisual redundancy:** Merupakan tipe redundansi yang timbul akibat persepsi visual. Pada persepsi visual manusia, informasi tertentu kurang penting, contohnya kuantisasi graylevel yang tepat tidak memengaruhi kualitas visualnya. Tipe redundansi ini dapat dikurangi dengan metode-metode pemampatan kuantisasi.

Ada 2 jenis metode kompresi untuk citra digital, yaitu:

- **Lossy Image Compression:** Merupakan tipe metode kompresi citra yang menghasilkan citra hasil pemampatan yang hampir sama dengan citra semula. Ada informasi yang hilang akibat pemampatan, tetapi dapat ditolerir oleh persepsi visual. Tipe metode kompresi citra ini bertujuan untuk memperoleh nisbah pemampatan yang tinggi. Contoh metode kompresi yang termasuk pada *Lossless Image Compression* diantaranya yaitu *Fractal Image Compression* dan *Discrete Cosine Transform (DCT) Compression, JPEG Compression*.
- **Lossless Image Compression:** Merupakan tipe metode kompresi citra yang selalu menghasilkan citra hasil peniripampatan yang tepat sama dengan citra semula, pixel per pixel. Citra hasil kompresi tidak memiliki informasi yang hilang akibat pemampatan. Selain itu, citra hasil kompresi juga memiliki nisbah pemampatan rendah, namun kualitas citra tetap tinggi. Tipe metode ini dibutuhkan untuk memampatkan citra yang tidak boleh terdegradasi akibat pemampatan, misalnya citra hasil rontgen / X-ray. Contoh metode yang termasuk dalam *Lossless Image Compression* diantaranya yaitu *Huffman Coding, Area Image Compression, Run-Length Encoding (RLE), dan Quantized Coding*.

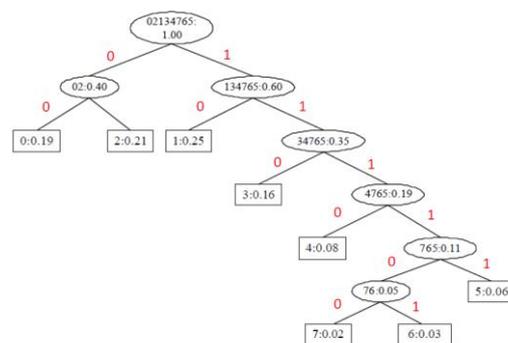


Gambar 7. Citra X-ray, salah satu contoh tipe citra yang membutuhkan *Lossless Image Compression* agar tidak terjadi kesalahan diagnosis pada pasien (Sumber: [X-Rays: MedlinePlus](#))

D. Huffman

Seperti yang telah disebutkan sebelumnya, Huffman Coding termasuk dalam metode *Lossless Image Compression*. Metode ini didasarkan pada algoritma *Greedy*. Prinsip kerjanya adalah pixel dengan nilai keabuan yang sering muncul dikodekan dengan panjang bit yang lebih sedikit, sebaliknya nilai keabuan yang jarang muncul dikodekan dengan bit yang lebih panjang. Langkah-langkah Huffman Coding adalah sebagai berikut:

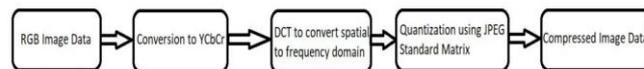
- Setiap nilai keabuan dinyatakan sebagai simpul. Setiap simpul di assign dengan frekuensi kemunculan nilai keabuan tersebut
- Urutkan secara menaik simpul simpul berdasarkan frekuensi kemunculannya.
- Gabung dua buah simpul yang mempunyai frekuensi kemunculan paling kecil pada sebuah akar. Akar mempunyai frekuensi yang merupakan jumlah dari frekuensi dua buah pohon penyusunnya.
- Ulangi langkah 2 sampai tersisa hanya satu buah pohon biner
- Beri label setiap sisi pada pohon biner. Sisi kiri dilabeli dengan 0 dan sisi kanan dilabeli dengan 1.
- Telusuri pohon biner dari akar ke daun. Barisan label-label sisi dari akar ke daun menyatakan kode Huffman untuk derajat keabuan yang bersesuaian



Gambar 8. Contoh pohon biner Huffman (Sumber: [25-Image-Compression-Bagian1-2022](#))

E. Discrete Cosine Transform Compression

Discrete Cosine Transform (DCT) merupakan tipe transformasi Fourier cepat yang memetakan sinyal riil ke nilai yang sesuai dalam domain frekuensi. DCT hanya bekerja pada bagian nyata dari sinyal kompleks karena sebagian besar sinyal dunia nyata adalah sinyal riil tanpa komponen kompleks.



Gambar 9. Langkah-langkah DCT Compression (Sumber: [Image Compression — DCT Method. DCT based Image Compression | by Ojaas Hampiholi | Towards Data Science](#))

Langkah-langkah implementasi DCT *Compression* adalah sebagai berikut:

- Konversi gambar RGB ke format YCbCr yang setara sebelum kita dapat melakukan pemrosesan DCT. Langkah penting lainnya di sini adalah mengubah rentang nilai piksel dari -128 menjadi 127, bukan 0 hingga 255.
- Pecah gambar menjadi blok berukuran N x N.
- Aplikasikan DCT ke setiap blok.
- Kuantisasi diterapkan untuk membatasi jumlah nilai yang dapat disimpan tanpa kehilangan informasi.
- Dekuantisasi gambar untuk melakukan dekompresi.

III. IMPLEMENTASI SOLUSI

Solusi dikembangkan menggunakan MATLAB R2022a. Program yang dibuat membutuhkan add-on bernama Image Processing Toolbox dan Signal Processing Toolbox. Penulis juga membuat MATLAB GUI App untuk program ini. Aplikasi akan menerima masukan sebuah citra berformat PNG untuk dimampatkan. Pengguna akan memilih metode kompresi yang tersedia, yaitu Huffman dan DCT. Pengguna juga dapat memasukkan nama berkas baru untuk citra hasil kompresi. Setelah diproses, aplikasi akan menampilkan citra awal dan citra hasil kompresi beserta ukuran masing-masing file. Nisbah hasil kompresi kemudian akan dicatat dan dianalisis kemudian.

A. Huffman Coding

Berikut ini merupakan implementasi fungsi untuk melakukan kompresi citra menggunakan *Huffman Coding*.

```
function res = huffmancoding(path, file)
    I = imread(fullfile(path, file));
    tempImg = I(:);
    [x, y, z] = size(I);
    [N, M] = size(tempImg);
    Count = zeros(256,1);
    for i = 1:N
        for j = 1:M
            Count(tempImg(i,j)+1)=Count(tempImg(i,j)+1)+1;
        end
    end
    prob = Count/(M*N);
    symbols = 0:255;
    %Huffman Dictionary
    [dict,avglen] =
    huffmandict(symbols,prob);
    %Huffman Encode
    code = huffmanenco(img,dict);
    %Huffman Decompression
    sig = huffmandeco(code, dict);
    %Reshape the 1-D array into 3-D array
    res = reshape(sig, [x, y, z]);
end
```

B. DCT Compression

Berikut ini adalah implemetansi fungsi untuk melakukan kompresi citra menggunakan DCT *Compression*.

```
function res = dctcoding(path, file)
    I = imread(fullfile(path, file));
    %Convert RGB into YCbCr
    YCbCr = rgb2ycbcr(I);
    Y = YCbCr(:,:,1);
    [h, w] = size(Y);
    r = h/8;
    c = w/8;
    s = 1;
    q50 = [16 11 10 16 24 40 51 61;
           12 12 14 19 26 58 60 55;
           14 13 16 24 40 57 69 56;
           14 17 22 29 51 87 80 62;
           18 22 37 56 68 109 103 77;
           24 35 55 64 81 104 113 92;
           49 64 78 87 103 121 120 101;
           72 92 95 98 112 100 103 99];
    % COMPRESSION
    for i=1:r
        e = 1;
        for j=1:c
            block = Y(s:s+7,e:e+7);
            cent = double(block) - 128;
            for m=1:8
                for n=1:8
                    if m == 1
                        u = 1/sqrt(8);
                    else
                        u = sqrt(2/8);
                    end
                    if n == 1
                        v = 1/sqrt(8);
                    else
                        v = sqrt(2/8);
                    end
                    comp = 0;
                    for x=1:8
                        for y=1:8
                            %DCT
                            comp = comp +
                            cent(x, y)*(cos((((2*(x-1))+1)*(m-1)*pi)/16))*cos((((2*(y-1))+1)*(n-1)*pi)/16));
                        end
                    end
                    F(m, n) = v*u*comp;
                end
            end
            for x=1:8
                for y=1:8
                    cq(x, y) = round(F(x,
                    y)/q50(x, y));
                end
            end
        end
    end
```

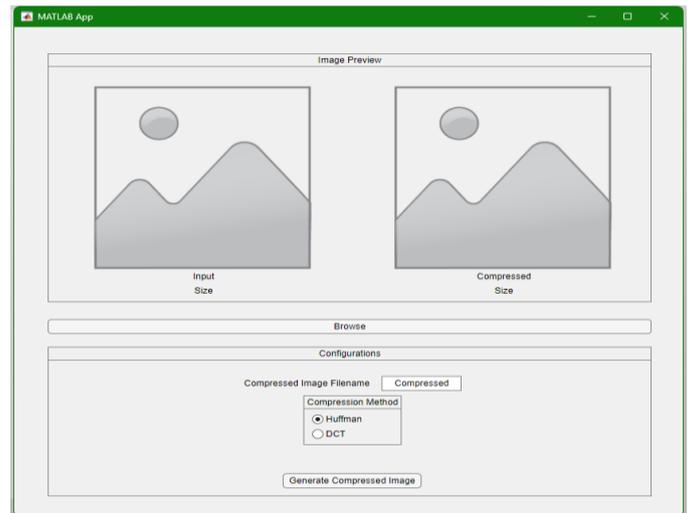
```

        end
        Q(s:s+7,e:e+7) = cq;
        e = e + 8;
    end
    s = s + 8;
end
s = 1;
for i=1:r
    e = 1;
    for j=1:c
        cq = Q(s:s+7,e:e+7);
        for x=1:8
            for y=1:8
                DQ(x, y) = q50(x,
y)*cq(x, y);
            end
        end
        for x = 1:8
            for y = 1:8
                comp = 0;
                for m = 1:8
                    for n = 1:8
                        if m == 1
                            u =
1/sqrt(2);
                        else
                            u = 1;
                        end
                        if n == 1
                            v =
1/sqrt(2);
                        else
                            v = 1;
                        end
                        %IDCT
                        comp = comp +
u*v*DQ(m, n)*(cos((((2*(x-1))+1)*(m-
1)*pi)/16))*(cos((((2*(y-1))+1)*(n-
1)*pi)/16));
                    end
                end
                bf(x, y) =
round((1/4) *comp + 128);
            end
        end
        Org(s:s+7,e:e+7) = bf;
        e = e + 8;
    end
    s = s + 8;
end
res = Y;
end

```

C. Graphical User Interface

Berikut merupakan tampilan GUI dari program MATLAB yang dibuat untuk makalah ini:



Gambar 10. GUI program MATLAB yang dibuat (Sumber: milik pribadi)

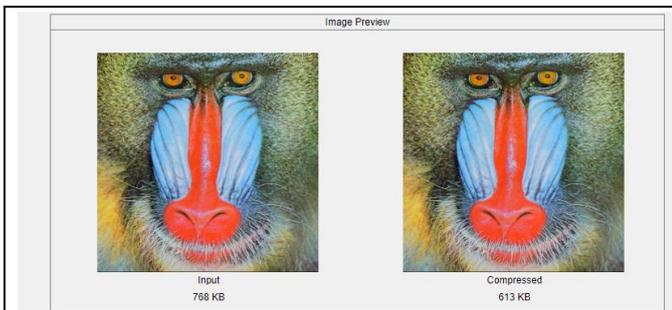
GUI dari program memiliki komponen sebagai berikut:

- 2 komponen gambar yang akan menampilkan citra masukan (kiri) dan citra hasil kompresi (kanan)
- Tombol untuk memilih citra masukan
- Edit Field untuk nama berkas citra hasil kompresi
- Tombol untuk memilih metode kompresi, yaitu Huffman dan DCT
- Label yang menampilkan ukuran berkas citra masukan dan citra hasil kompresi
- Tombol untuk membuat citra kompresi

IV. HASIL & ANALISIS

A. Hasil

Tabel berikut ini merupakan hasil eksekusi program menggunakan 5 citra berbeda dan masing-masing dikompres dengan metode *Huffman Coding* dan *DCT Compression*.



Browse

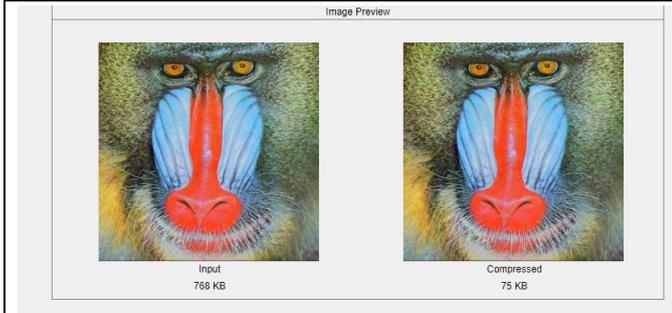
Configurations

Compressed Image Filename

Compression Method

Huffman

DCT



Browse

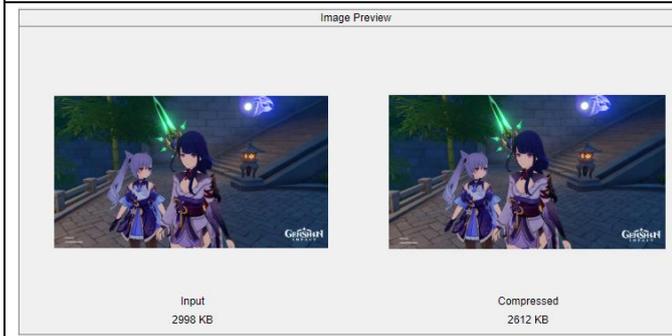
Configurations

Compressed Image Filename

Compression Method

Huffman

DCT



Browse

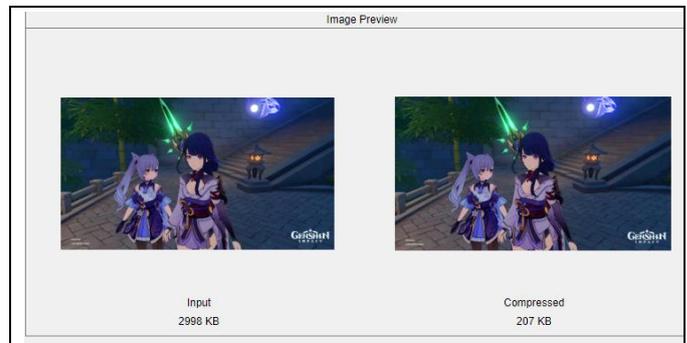
Configurations

Compressed Image Filename

Compression Method

Huffman

DCT



Browse

Configurations

Compressed Image Filename

Compression Method

Huffman

DCT



Browse

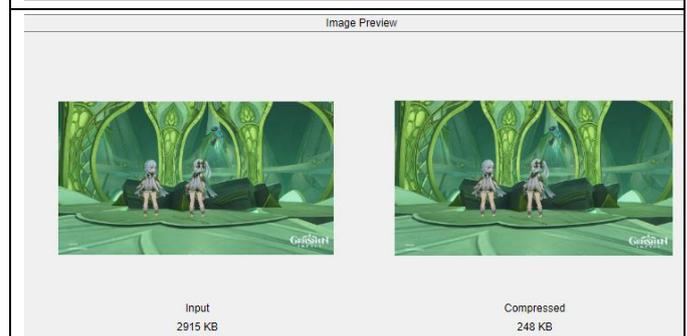
Configurations

Compressed Image Filename

Compression Method

Huffman

DCT



Browse

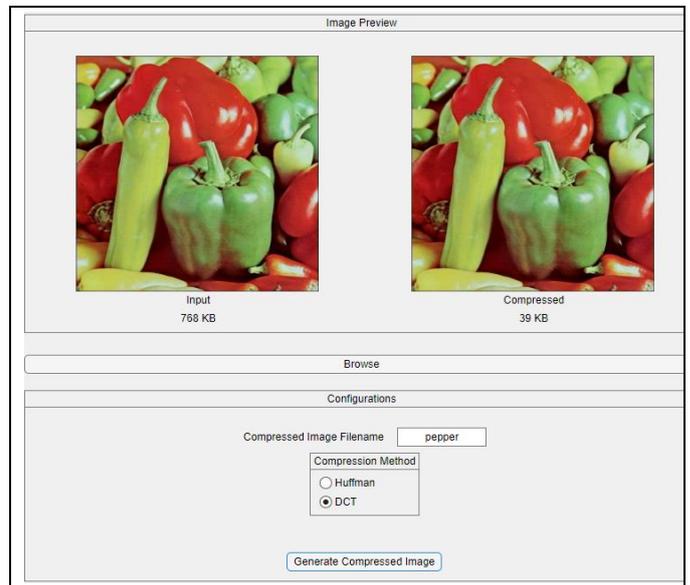
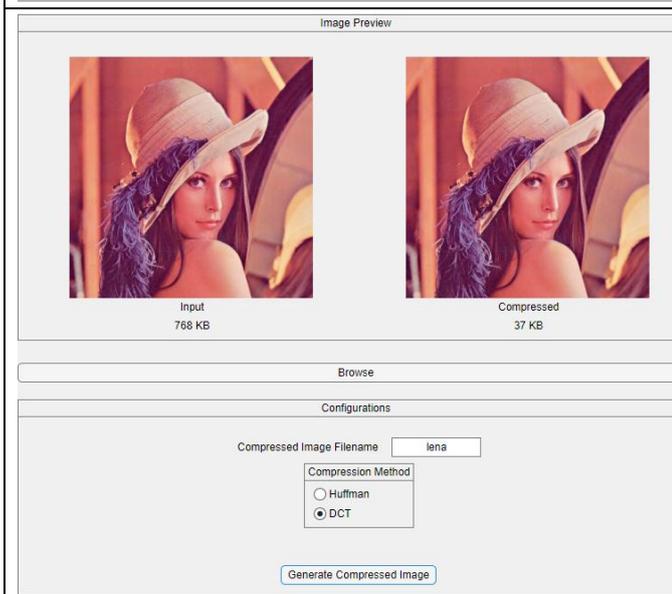
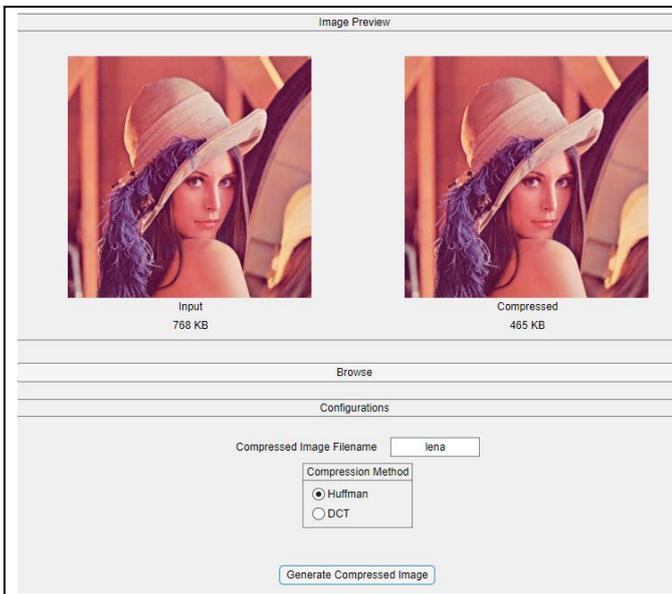
Configurations

Compressed Image Filename

Compression Method

Huffman

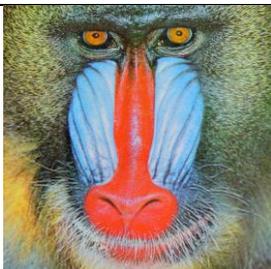
DCT

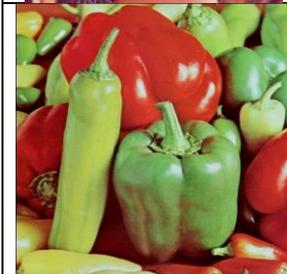


Tabel 1. Hasil eksekusi program di 5 citra berbeda

B. Analisis

Tabel berikut adalah perbandingan nisbah untuk masing-masing citra:

| Citra | Metode | Ukuran | Nisbah |
|--|---------|---------------------------------------|--------|
|  | Huffman | Awal: 768 KB Akhir: 613 KB | 79.82% |
| | DCT | Awal: 768 KB Akhir: 75 KB | 9.77% |
|  | Huffman | Awal: 2998 KB Akhir: 2612 KB | 87.12% |
| | DCT | Awal: 2998 KB Akhir: 207 KB | 6.9% |
|  | Huffman | Awal: 2915 KB Akhir: 2547 KB | 87.38% |
| | DCT | Awal: 2915 KB Akhir: 248 KB | 8.51% |

| | | | |
|--|---------|-------------------------------------|--------|
|  | Huffman | Awal: 768 KB Akhir: 465 KB | 60.55% |
| | DCT | Awal: 768 KB Akhir: 37 KB | 4.82% |
|  | Huffman | Awal: 768 KB Akhir: 417 KB | 54.3% |
| | DCT | Awal: 768 KB Akhir: 39 KB | 5.08% |

Tabel 2. Perhitungan nisbah untuk masing-masing citra pada kompresi citra menggunakan metode DCT dan Huffman.

Dapat dilihat pada tabel 2, dapat dilihat bahwa nisbah kompresi kelima citra menggunakan *Huffman Coding*, yang merupakan *Lossless Image Compression*, memiliki rata-rata **73.83%**. Sementara itu, jika menggunakan *DCT Compression* yang merupakan *Lossy Image Compression*, rata-rata nisbah kompresi kelima citra adalah **7.02%**.

V. KESIMPULAN

Metode *Huffman Coding* dan *DCT Compression* berhasil mengurangi ukuran berkas citra, meski dengan skala yang berbeda. Keduanya cocok digunakan untuk pembuatan keluku. Namun, penggunaan kedua metode ini untuk pembuatan keluku tetap harus memerhatikan tujuan, apakah Anda membutuhkan citra itu saja ataukah Anda juga membutuhkan semua informasi yang terkandung dalam berkas citra. Jika Anda membutuhkan berkas dengan ukuran seminimum mungkin, tentunya Anda dapat menggunakan *DCT Compression* ataupun metode *Lossy Image Compression* lainnya yang ada.

VI. UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Tuhan Yang Maha Esa, karena berkat kehadiran-Nya, penulis dapat menyelesaikan makalah ini. Tak lupa penulis berterima kasih kepada kedua orang tua penulis, serta Bapak Dr. Ir. Rinaldi, MT sebagai dosen mata kuliah Interpretasi dan Pengolahan Citra. Terakhir, penulis ingin berterima kasih kepada sahabat penulis yang bersedia membantu menyumbangkan ide selama pembuatan makalah ini.

REFERENCES.

- [1] <https://informatika.stei.itb.ac.id/~rinaldi.munir/Citra/2022-2023/01-Pengantar-Pengolahan-Citra-Bag1-2022.pdf> (Diakses pada Desember 2022)
- [2] <https://informatika.stei.itb.ac.id/~rinaldi.munir/Citra/2021-2022/16-Warna-bagian1-2022.pdf> (Diakses pada Desember 2022).
- [3] <https://informatika.stei.itb.ac.id/~rinaldi.munir/Citra/2022-2023/25-Image-Compression-Bagian1-2022.pdf> (Diakses pada Desember 2022)
- [4] [What is a thumbnail and how is it used? \(techtargget.com\)](https://techtargget.com/what-is-a-thumbnail-and-how-is-it-used/) (Diakses Desember 2022)
- [5] <https://medlineplus.gov/xrays.html> (Diakses pada Desember 2022)
- [6] <https://towardsdatascience.com/image-compression-dct-method-f2bb79419587> (Diakses pada Desember 2022)

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, 19 Desember 2022



Wisnu Aditya Samiadji - 13519093