

Implementasi Pemampatan Beruntun *Hierarchical Clustering*, *Run Length Encoding*, dan *Huffman*

Kinantan Arya Bagaspati / 13519044
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
13519044@gmail.com

Abstract—Pemampatan citra merupakan hal yang tidak dapat dipungkiri ada dalam keseharian tanpa disadari khalayak umum seperti pada penyimpanan gambar dalam format file tertentu. Diantara banyak metode pemampatan, ternyata terdapat kombinasi pemampatan yang ditelusuri dalam makalah ini yang dapat memberi nilai artistik pada langkah demi langkahnya terhadap tren gambar minimalis. Metode ini ialah *hierarchical clustering* diikuti RLE dan *Huffman* sebagai pemampatnya.

Keywords—*citra, pemampatan, clustering*

I. PENDAHULUAN

Jika ditanya mengenai sebuah ucapan mengenai gambar atau citra, perkataan satu citra dapat bermakna seribu kata mungkin muncul dalam benak. Sama halnya dengan teks, citra merupakan salah satu representasi data yang pastinya membawa informasi. Representasi ini dapat divisualisasikan umumnya ke spasial 2 dimensi, yang secara kasat mata berupa kombinasi warna yang disajikan dalam bentuk berbeda dalam suatu orientasi tertentu.

Sebagai bentuk representasi data, citra merupakan salah satu objek yang dapat ditelaah dan ditelusuri lebih lanjut dalam bidang keilmuan Informatika. Sebutan yang sering muncul dalam pembahasan yang berkaitan dengan citra adalah *Computer Vision*. Cakupan dari *computer vision* ini sangat luas dari representasi proyeksi lingkungan tiga dimensi yang dinamis ke dalam layar yang terbatas dua dimensi, aplikasi teknik berupa efek yang dapat direkayasa komputer terhadap suatu media, metode untuk *rendering*, simulasi, dan animasi, dan pengolahan berbagai bentuk dan metode yang dapat diaplikasikan dalam suatu citra atau animasi. Dapat disimpulkan pengolahan citra merupakan salah satu cabang dari *computer vision* yang paling relevan dalam pembahasan karya tulis ini.

Sekali lagi sebagai bentuk informasi, kadangkala representasi data paling naif tidak berarti paling efisien mewakili informasi dalam segi memori maupun waktu. Kompresi atau pemampatan terhadap suatu citra umumnya merupakan salah satu bentuk pengurangan memori yang dibutuhkan untuk mewakili informasi dalam derajat yang sama. Derajat disini berarti setiap metode pemampatan tidak harus dapat secara penuh dikembalikan ke bentuknya semula.

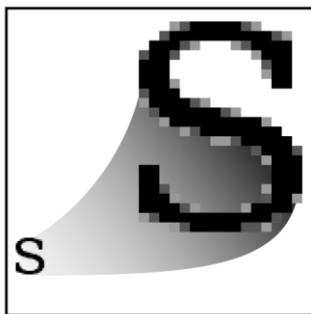
Pemampatan yang dapat dikembalikan secara sempurna berarti tidak mengurangi informasi sama sekali dalam pemampatan, biasa disebut *lossless compression*, sebaliknya merupakan *lossy compression*. *Lossy compression* mengorbankan sejumlah informasi yang dianggap tidak terlalu representatif akan citra keseluruhan, untuk mencapai rasio kemampatan yang umumnya jauh lebih tinggi ketimbang *lossless compression*. Pembagian ini tidak menutup kemungkinan penggabungan metode pemampatan dari *lossless* dan *lossy* untuk saling mengatasi kekurangannya masing-masing. Pada karya tulis ini dipilih metode *lossy compression* yang terinspirasi dari *hierarchical clustering* dan akan menghasilkan citra yang terbagi-bagi menjadi sebuah kelompok satu warna. Sifat lokalitas ini, yakni bagian berwarna sama cenderung terletak pada bagian sama, kemudian diaplikasikan *Run Length Encoding* (RLE) yang tentunya memanfaatkan fakta lokalitas tersebut lebih baik dari encoding lainnya.

Aplikasi pemampatan sangat banyak digunakan pada pengolahan citra pada kehidupan sehari-hari. Perubahan format gambar sangat umum menggunakan *lossy compression* untuk penghematan memorinya. Pengorbanan informasi dianggap wajar dalam pengolahan citra, dikarenakan fungsi alami citra untuk diapresiasi secara visual oleh konsumennya yakni manusia. Alat visual manusia lebih cenderung mengandalkan pengenalan pola ketimbang prinsip lain, sehingga dapat seolah-olah mengisi bagian seharusnya dari informasi citra yang hilang. Sebagai objek visual manusia, citra seringkali membawa unsur artistik tanpa disadari meski bukan fungsi utamanya dalam kasus itu. Penggunaan *lossless compression* juga cocok dengan tumbuhnya salah satu tren yang mengaplikasikan prinsip nilai artistik "*less is more*" dari suatu citra, yakni citra minimalis seperti dicontohkan pada gambar di bawah.



II. LANDASAN TEORI

A. Representasi Citra Digital



Raster
GIF, JPEG, PNG

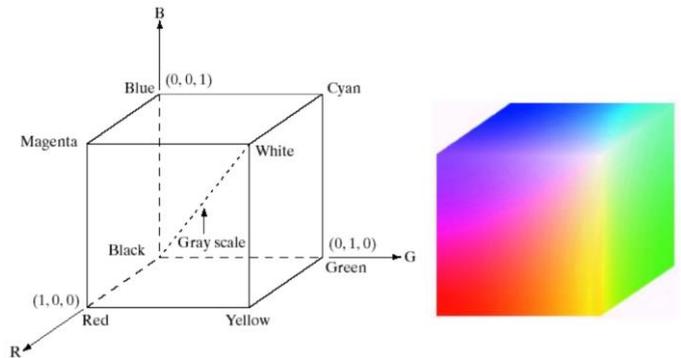


Vector
SVG

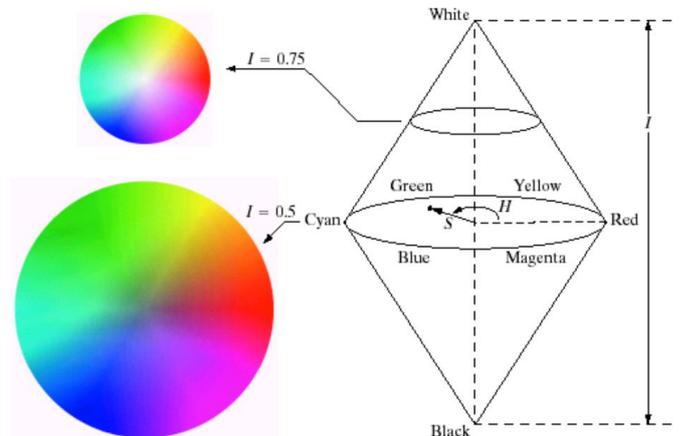
Dapat disimpulkan dari paparan sebelumnya bahwa terdapat banyak bentuk representasi data yang dapat digunakan untuk mewakili informasi pada suatu citra. Namun umumnya terdapat 2 aliran yang dapat mengelompokkan hampir seluruh representasi citra, yakni *pixel based* dan *vector based*. Citra yang direpresentasikan dalam pixel pada dasarnya menyatakan dirinya dalam tabel yang memiliki berhingga jumlah kolom dan baris dari petak-petak persegi yang dianggap satuan terkecil yang tidak dapat dibagi dalam citra yang disebut pixel. Sementara memanfaatkan sifat bahwa citra merupakan susunan bentuk-bentuk dengan warna masing-masing dengan border atau tepi yang umumnya berbentuk torehan garis atau lekukan, representasi vector pada dasarnya menyimpan informasi tepi yang membatasi tiap dua bagian berbeda warna pada citra, sehingga lebih fleksibel dalam proses skalar ketimbang pixel karena tidak memiliki satuan terkecil.

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

Saat ini, representasi pixel masih dihitung lebih marak dalam dunia sehari-hari. Oleh karena itu metode pemampatan pada karya tulis ini tentunya akan lebih terfokus pada yang dapat diaplikasikan dalam citra yang direpresentasikan secara pixel. Seperti gambar di atas, citra direpresentasikan sebagai matrix yang setiap sel melambangkan satu pixel. Setiap pixel mengandung setiap informasi warna yang seharusnya diperlihatkan pada posisi itu. Sehingga dapat disimpulkan tiap pixel mempunyai property urutan baris, kolom, dan representasi warna yang selanjutnya akan dibahas secara singkat.



Terdapat banyak metode representasi warna. Angka yang digunakan untuk menyatakannya biasa berkisar antara 0 hingga 255 (uint8). Jika terdapat 1 dimensi saja, umumnya angka tersebut menyatakan derajat keabuan atau grayscale, sebagai salah satu metode paling dasar dalam menyatakan warna dengan 0 sebagai hitam dan 255 sebagai putih. Selain itu, sebagian besar metode representasi warna menggunakan tiga dimensi, salah satu yang paling umum digunakan ialah RGB. Setiap angka berkisar 0-255 yang menyatakan seberapa merah, seberapa hijau, dan seberapa biru pixel ini. Hitam pada RGB bernilai (0,0,0) sementara putih bernilai (255, 255, 255). Metode RGB dan lawannya CMY (*Cyan Magenta Yellow*) didasarkan pada representasi komputer atau tinta printer.



Selain itu terdapat dekomposisi tiga dimensi warna yang lebih intuitif terhadap alat visualisasi mata manusia, yakni HSI. Hue menyatakan derajat warna yang berupa sirkular diawali merah, bergradasi ke hijau, ke biru, dan kembali lagi ke merah. Saturation melambangkan pudar jika 0 atau memiliki warna yang kuat (semakin jauh dari gray) jika mendekati 255. Terakhir Intensity menyatakan keterangan warna, gelap jika 0,

semakin putih/terang jika 255. Metode HSV dapat dibidang serupa dengan HSI, selain itu masih banyak representasi warna lainnya.

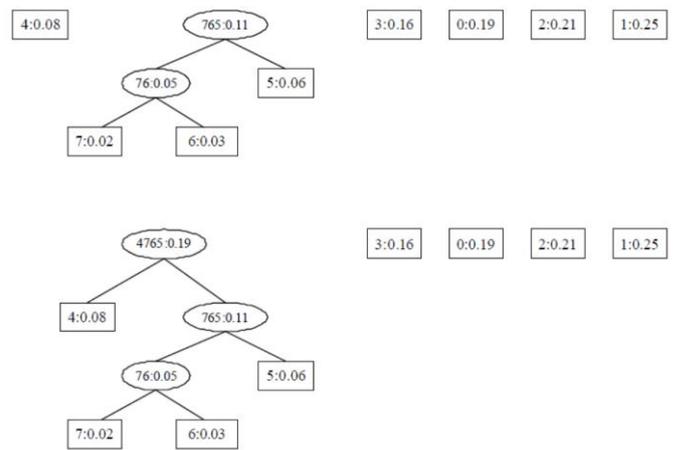
B. Pemampatan yang Lossless Compression

Telah dipaparkan bahwa lossless compression merupakan pengurangan jumlah memori yang dibutuhkan untuk menyatakan informasi yang serupa pada citra awal. Terdapat dua metode pemampatan yang akan dibahas dan digunakan dalam karya tulis ini yakni *Run Length Encoding* (RLE) dan *Huffman*. Akan dibahas metode masing-masing pemampatan dan kasus dimana tiap metode paling diuntungkan dalam memampatkan citra tersebut.

Sesuai namanya, metode RLE dapat dilakukan selagi menelusuri tiap pixel pada citra, ditambah lagi sangat sederhana dan simpel logikanya. Dalam penelusuran nilai tiap pixel, setiap nilai sama yang berurutan akan disingkat dalam bentuk nilai tersebut dan banyaknya pengulangan. Misalnya terdapat citra 3x3 yang bila ditelusuri memiliki nilai 111000022, maka RLE akan menyimpan informasi 3 buah 1, 4 buah 0, diikuti 2 buah 2. Logika sederhana ini terbukti dapat memampatkan secara signifikan apabila banyak pixel berdekatan yang bernilai sama, seperti pada gambar biner (hanya memiliki 2 nilai yang menandakan hitam atau putih). Oleh karena itu metode ini sering dipasangkan pada metode *lossy* yang cenderung menyamakan nilai pixel-pixel yang berdekatan, seperti metode clustering yang akan dibahas pada karya tulis ini.

Selanjutnya metode Huffman sebetulnya cukup umum dalam memampatkan, tidak hanya pada pengolahan citra. Metode Huffman menggunakan algoritma yang sebagian besar menganggapnya bersifat heuristik atau *greedy* untuk mencari konfigurasi pernyataan setiap nilai yang muncul dalam bit sedemikian sehingga bila dijejerkan, representasi bit tidak ambigu (merepresentasikan tepat satu citra awal), serta banyak bit merupakan global minimal. Berikut langkah-langkah menemukan representasi bit tiap nilai yang muncul dengan metode Huffman

1. Catat frekuensi tiap nilai yang muncul pada data.
2. Buat antrian dengan prioritas diurutkan dari frekuensi terendah.
3. Pilih 2 item dengan prioritas terendah pada antrian, gabungkan 2 item tersebut dengan label jika awalnya a dan b, menjadi (a)(b). Prioritas item baru ini merupakan jumlah prioritas 2 item pembentuknya, lalu masukkan lagi dalam antrian prioritas.
4. Lakukan 3 terus hingga antrian berisi 1 item
5. String yang dibentuk dapat direpresentasikan dalam pohon biner mengikuti tanda kurung yang telah ditambahkan seiring tiap langkah penggabungan.
6. Setiap nilai pada data awal merupakan daun dari pohon biner ini. Pasangkanlah setiap pembagian pada simpul yang bukan daun pada pohon dengan bit 0 dan 1, maka representasi biner tiap daun ialah konkatenasi tiap bit dari jalur terurut dari akar pohon hingga daun tersebut.

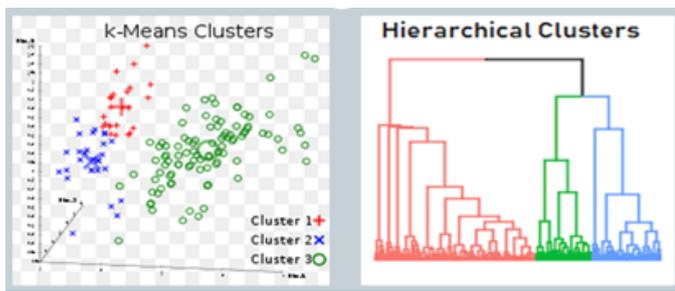


Gambar di atas merupakan contoh penggabungan 2 item pertama dengan frekuensi berturut-turut 8 dan 11 menjadi entitas pohon biner dengan prioritas 19, sebelum dimasukkan pada antrian prioritas lagi.

C. Hierarchical Clustering

Clustering sesuai namanya merupakan metode mengelompokkan sejumlah bentuk data berdasarkan properti atau karakteristiknya, sedemikian sehingga memaksimalkan kedekatan setiap pasang data dalam kelompok yang sama, dan memaksimalkan jumlah jarak setiap pasang data dari kelompok yang berbeda. Clustering dapat dibidang merupakan konsep cukup asing dalam pemampatan citra karena berangkat dari bidang keilmuan ilmu data dan pembelajaran mesin. Meskipun demikian clustering tentunya tidak asing dalam klasifikasi citra, karena merupakan penurunan dari fungsi alaminya.

Secara garis besar, clustering dibagi menjadi 2 kelompok besar diantaranya *partitional clustering* dan *hierarchical clustering*. *Partitional clustering* ialah pengelompokan berdasarkan pembagian yang terlebih dahulu telah ditetapkan jumlahnya. Biasanya clustering ini memiliki benih awal setiap kelompok, yang menentukan tiap data lain masuk ke dalam kelompok dengan benih apa. Salah satu diantaranya Kmeans yang menghitung ulang benih tiap kelompok sebagai nilai rata-ratanya, dan lainnya seperti Kmedoids. Di lain sisi, prosedur *hierarchical clustering* berawal dari tiap data merupakan clusternya masing-masing, hingga semua data menjadi satu kelompok. Setiap langkahnya terjadi penggabungan antar dua kelompok yang menurut algoritma yang digunakan paling menguntungkan jika dilihat dari sisi objektif. Biasanya pada pembelajaran mesin salah satu keadaan antara awal hingga akhir proses tersebut terdapat pembagian klustering yang ideal. Kita akan berfokus pada *partitional clustering*.



Pada suatu metode hierarchial clustering, terdapat beberapa komponen pokok yang dibutuhkan untuk menyatakan algoritmanya

1. Metode penghitungan jarak sepasang data. Jarak yang umum digunakan misalnya Euclidean, Manhattan, dot product untuk mencari cosinus dari sudut yang dibentuk vector, dan lain sebagainya.
2. Metode penghitungan jarak sepasang kluster untuk menentukan 2 kluster mana yang berjarak minimal untuk digabung. Biasanya dipilih salah satu dari 3 metode yakni jarak terdekat, jarak rata-rata, atau jarak terjauh antara sepasang data berbeda kluster tersebut.
3. Silhouette coefficient sebagai metrik yang menentukan keefektifan separasi kluster yang telah dihasilkan berdasarkan rata-rata jarak sepasang data antar kluster dan inter kluster.

D. Evaluasi Pemampatan

Hasil pemampatan dari suatu metode dapat dilakukan evaluasi sebagai pembandingan kinerja metode pemampatan yang satu dengan yang lainnya. Umumnya terdapat 2 dimensi atau sudut pandang dilakukannya evaluasi pada pemampatan yakni terkait memori, serta informasi yang hilang terhadap citra awal. Terkait memori dapat digunakan rasio pemampatan, sedangkan terkait informasi yang dikorbankan dapat digunakan pembandingan dengan citra awal, salah satunya PSNR.

We suggest that you use a text box to insert a graphic (which is ideally a 300 dpi resolution TIFF or EPS file with all fonts embedded) because this method is somewhat more stable than directly inserting a picture.

To have non-visible rules on your frame, use the MSWord "Format" pull-down menu, select Text Box > Colors and Lines to choose No Fill and No Line.

Rasio pemampatan atau disebut nisbah sebetulnya cukup sederhana logikanya, yakni sebatas rasio memori yang dibutuhkan dalam citra hasil pemampatan dibandingkan dengan citra lama. Semakin kecil nisbah ini berarti semakin hemat memori citra yang dihasilkan

$$Nisbah1 = \frac{\text{ukuran citra sesudah dimampatkan}}{\text{ukuran citra sebelum dimampatkan}} \times 100\%$$

Selanjutnya PSNR (*peak signal to noise ratio*) merupakan metrik yang menandakan tingkat perbedaan citra awal dengan citra hasil, berdasarkan formula yang diaplikasikan pada

perbandingan setiap pixelnya. PSNR dihitung dalam satuan desibel dengan formula lengkap

$$PSNR = 20 \times \log_{10} \left(\frac{b}{rms} \right)$$

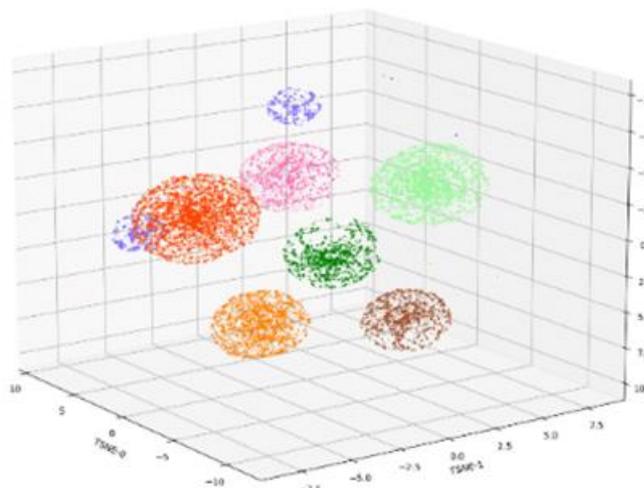
$$rms = \sqrt{\frac{1}{\text{Lebar} \times \text{Tinggi}} \sum_{i=1}^N \sum_{j=1}^M (f_{ij} - f'_{ij})^2}$$

Semakin tinggi PSNR, semakin mirip dua citra yang dibandingkan.

III. IMPLEMENTASI

Alangkah baiknya bila dipaparkan langkah-langkah implementasi yang runtut dari penggunaan teori-teori ini dalam suatu citra.

1. Citra yang direpresentasikan dalam bentuk pixel dapat divisualisasikan dalam penggabungan dimensi spasial lokasi pixel dengan nilai warna pixel tersebut. Sebagai contoh jika citra berupa grayscale maka hanya ada 1 dimensi warna sehingga total 3 dimensi dapat divisualisasikan dengan berbagai perangkat yang tersedia, salah satunya python seperti gambar di bawah.



2. Terlebih dahulu sebelum klustering, jika diperlukan, dilakukan scaling yang perlu dieksperimentasikan untuk setiap gambar karena mempengaruhi hasil intermediate dari pertengahan proses hierarchial clustering. Scaling dilakukan untuk mengatur skala pengaruh setiap dimensi terhadap penghitungan jarak. Sebagai penghitungan dasar, karya tulis ini menggunakan scaling ke 256 interval untuk semua dimensi
3. Kumpulan data yang telah berkisar antara nilai yang diinginkan kemudian siap untuk dilakukan hierarchial

klustering. Setiap kluster memiliki nilai rata-rata yang mewakili nilai data-data pada kelompok tersebut. Penggabungan pada tiap langkah akan memanfaatkan struktur data *disjoint set union* (DSU) untuk menjaga koefisien pengali pada kompleksitas waktu tetap log.

4. Pada awal dibuat antrian prioritas yang berisi pasangan kluster, dengan jarak tiap pasang rata-rata kluster sebagai nilai prioritasnya. Jika terdapat N buah pixel, maka tahap ini memiliki kompleksitas $O(N^2 \log(N))$
5. Ambil elemen dengan nilai prioritas terkecil, tandanya pasangan pada elemen ini ialah 2 kluster yang digabungkan. Menurut prinsip DSU, kluster yang lebih kecil menjadi keturunan kluster yang lebih besar.
6. Nilai rata-rata pada kluster yang lebih besar kemudian diganti sesuai rata-rata baru, dan tentunya menghapus setiap entri pada antrian prioritas yang terpengaruh. Terdapat $O(N)$ entri yang terpengaruh sehingga perubahan antrian pada langkah ini membutuhkan $O(N \log(N))$
7. Langkah 6 diulang hingga antrian habis, yakni N iterasi, maka total kompleksitas pengulangan langkah 6 ialah $O(N^2 \log(N))$
8. Untuk setiap pencetakan citra, dalam menelusuri keturunan hingga teratas, setiap pixel membutuhkan $O(\log(N))$ iterasi menurut DSU, untuk mengetahui nilai rata-rata kluster yang ditinggalinya. Oleh karena itu penggambaran memiliki kompleksitas $O(N \log(N))$ ketimbang $O(N)$.
9. Untuk setiap pencetakan citra, dapat dilakukan pemampatan menggunakan RLE kemudian Huffman, jika terdapat M nilai berbeda yang akan dilakukan metode Huffman, pastinya memiliki kompleksitas $O(\log M)$ dalam setiap langkah penggabungannya, sehingga kompleksitas waktu maupun memori (jumlah bit) dapat disimpulkan sebesar $O(M \log M)$. Perlu diperhatikan bahwa M tidak mungkin lebih dari N .

Setelah diamati, setiap langkahnya nilai PSNR akan menurun karena citra yang dihasilkan semakin mengorbankan informasi, bisa dibilang semakin minimalis. Namun di sisi lain nilai rasio kemampuan menaik dalam setiap iterasi, dikarenakan penggabungan kluster membuat RLE dan Huffman semakin efisien dan mampat. Dari kedua grafik yang berbanding terbalik, tetap dapat ditarik kesimpulan untuk mendapatkan *elbow point*, yakni iterasi yang menandakan klustering terbaik dari perubahan laju penurunan PSNR atau kenaikan rasio kemampuan. Kesimpulan lain yang dapat diambil juga ialah kompleksitas waktu total keseluruhan langkah apabila dilakukan analisis kemampuan dan pencetakan gambar tiap iterasi penggabungan kluster, ialah $O(N^2 \log(N))$

IV. KESIMPULAN DAN SARAN

Sesuai dengan pernyataan awal, pengaplikasian metode pemampatan lossless dapat digunakan tepat setelah dilakukan pemampatan atau setidaknya transformasi yang bersifat *lossy* hal ini tercerminkan dari penggunaan RLE dan kemudian Huffman setelah citra diaplikasikan transformasi setelah selesai dikelompokkan menggunakan hierarchial clustering. Properti-proerti lokalitas ini menguntungkan diaplikasikan RLE yang sangat meningkatkan performa kemampuan bila diperhatikan dari metrik nisbahnya. Hal ini tentunya tidak menutup kemungkinan lain adanya transformasi lain yang cocok diaplikasikan RLE, serta metode pemampatan juga dapat berupa metode lain selain RLE. Manfaat clustering juga dapat dirasakan dalam tren minimalis terhadap suatu citra, ditambah lagi banyaknya hasil yang dapat dipilih dari proses iterasi tergantung banyaknya jumlah kluster.

Seiring berjalannya perkembangan teknologi, dapat saja terjadi penggantian tren sedemikian sehingga representasi citra berbasis vector lebih populer dan akan menjadi bentuk yang lebih dasar atau umum ketimbang vector. Oleh karena itu dapat dianalisa viabilitas metode ini apabila diaplikasikan pada ranah representasi vector ketimbang biner. Hal yang dapat dispekulasikan ialah RLE yang dapat digantikan dengan metode lain yang memanfaatkan sejumlah vector pembatas daerah setiap kluster. Bagian lain yang dapat diperbaharui ialah fitur untuk memodifikasi kluster tertentu secara manual maupun waktu saat di tengah-tengah iterasi *hierarchial clustering*. Modifikasi ini dapat berupa penggabungan secara manual, penggantian warna, dan lainnya yang sejalan dengan maksud *clustering* memberi kebebasan pada artis dalam membuat karya minimalis.

UCAPAN TERIMA KASIH

Pertama-tama, penulis mengucapkan terimakasih kepada Tuhan Yang Maha Esa atas berkat dan rahmatnya, penulis bisa menyelesaikan tugas makalah ini. Penulis juga mengucapkan terimakasih kepada Bapak Rinaldi Munir selaku dosen mata kuliah Interpretasi dan Pengolahan Citra, yang selama ini membimbing pembelajaran Interpretasi dan Pengolahan Citra yang sangat membantu pengerjaan makalah ini, sekaligus menyediakan situs yang dapat dengan mudah diakses berisi materi-materi kuliah, latihan-latihan soal untuk kuis dan ujian, dan semua dokumen pembelajaran, soal, dan lainnya yang tentunya berguna dalam proses pembelajaran Interpretasi dan Pengolahan Citra.

REFERENSI

- [1] <https://informatika.stei.itb.ac.id/~rinaldi.munir/Citra/2022-2023/01-Pengantar-Pengolahan-Citra-Bag1-2022.pdf>, diakses pada 19 Desember 2022
- [2] <https://informatika.stei.itb.ac.id/~rinaldi.munir/Citra/2021-2022/23-Segmentasi-Citra-Bagian2-2022.pdf>, diakses pada 19 Desember 2022
- [3] <https://informatika.stei.itb.ac.id/~rinaldi.munir/Citra/2022-2023/25-Image-Compression-Bagian1-2022.pdf>, diakses pada 19 Desember 2022
- [4] <https://informatika.stei.itb.ac.id/~rinaldi.munir/Citra/2021-2022/16-Warna-bagian1-2022.pdf>, diakses pada 19 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

A handwritten signature in black ink, appearing to be 'Kinantan Arya Bagaspati', written in a cursive style.

Kinantan Arya Bagaspati