

Kombinasi Metode (RLE) dengan Algoritma Arithmetic Coding dalam kompresi file citra

Ferdian Sevrizal Gowasa¹, Aprianna Simanjuntak^{2*}

¹ Fakultas Ilmu Komputer, Program Studi Sistem Informasi, Universitas Audi Indonesia, Medan, Indonesia

² Fakultas Ilmu Komputer, Program Studi Sistem Informasi, Universitas Audi Indonesia, Medan, Indonesia
Email: 1ferdiansevrizal06@gmail.com ² apriannasimanjuntak0@gmail.com*

Abstrak—Saat kita mengirimkan citra digital dengan menggunakan jalur komunikasi atau internet. Dengan ukuran *file* yang besar maka menimbulkan masalah pada pengiriman gambar yaitu waktu pengiriman yang lama. Masalah lain adalah semua gambar memerlukan media penyimpanan (*storage*) yang cukup besar. Hal ini dapat menimbulkan masalah yang cukup serius ketika gambar disimpan dalam *database* dengan keterbatasan media penyimpanan yang ada. Sehingga diupayakan suatu solusi yang dapat mereduksi besarnya ukuran file gambar. Salah satu solusi untuk mengatasi masalah di atas adalah dengan melakukan kompresi. Kompresi gambar adalah proses untuk meminimalisasi jumlah bit yang mempresentasikan suatu gambar sehingga ukuran file menjadi lebih kecil. Ada dua tipe kompresi yaitu tipe *lossless* dan kompresi tipe *lossy*. Kompresi tipe *lossless* adalah kompresi dimana kualitas hasil kompresi tidak menurun ketika direkonstruksi. Sedangkan tipe *lossy* akan menghasilkan kualitas gambar yang dihasilkan jauh lebih rendah dari pada kualitas citra asli ketika direkonstruksi.

Kata Kunci: Kompresi, Run Length Encoding, Aritmetic Coding

Abstract—When we send digital images using communication lines or the internet. With a large file size, it causes problems with sending images, namely long delivery times. Another problem is that all images require quite large storage media. This can cause quite serious problems when images are stored in a database with limited storage media. So we are trying to find a solution that can reduce the size of image files. One of the solutions to overcome the problem above is to perform compression. Image compression is the general process of minimizing the number of bits that represent an image so that the file size becomes smaller. There are two types of compression, namely lossless type and lossy type compression. Lossless type compression is compression where the quality of the compression results does not decrease when reconstructed. Meanwhile, the lossy type will result in the resulting image quality being much lower than the quality of the original image when reconstructed.

Keywords: *Compression, Run Length Encoding, Arithmetic Coding*

1. PENDAHULUAN

Saat kita mengirimkan citra digital dengan menggunakan jalur komunikasi atau internet. Dengan ukuran *file* yang besar maka menimbulkan masalah pada pengiriman gambar yaitu waktu pengiriman yang lama[1]. Masalah lain adalah semua gambar memerlukan media penyimpanan (*storage*) yang cukup besar. Hal ini dapat menimbulkan masalah yang cukup serius ketika gambar disimpan dalam *database* dengan keterbatasan media penyimpanan yang ada. Sehingga diupayakan suatu solusi yang dapat mereduksi besarnya ukuran file gambar. Salah satu solusi untuk mengatasi masalah di atas adalah dengan melakukan kompresi[2].

Kompresi gambar adalah proses untuk meminimalisasi jumlah bit yang mempresentasikan suatu gambar sehingga ukuran file menjadi lebih kecil. Ada dua tipe kompresi yaitu tipe *lossless* dan kompresi tipe *lossy*. Kompresi tipe *lossless* adalah kompresi dimana kualitas hasil kompresi tidak menurun ketika direkonstruksi. Sedangkan tipe *lossy* akan menghasilkan kualitas gambar yang dihasilkan jauh lebih rendah dari pada kualitas citra asli ketika direkonstruksi[3][4].

Pada penelitian ini, penulis membahas tentang kompresi citra menggunakan dua metode yaitu metode *Run Length Encoding* (RLE) Dan algoritma *Aritmetic Coding*. Pada kompresi pertama digunakan metode *Run Length Encoding*, dan *output* hasil kompresi metode *Run Length Encoding* dijadikan data untuk kompresi selanjutnya yang menggunakan algoritma *arithmetic coding*, dengan tujuan untuk mengetahui apakah hasil kompresi menjadi *lossless compression* atau *lossy compression* karena *Arithmetic Coding* lebih banyak diaplikasikan pada *file-file* dokumen umum, sedangkan *Run*

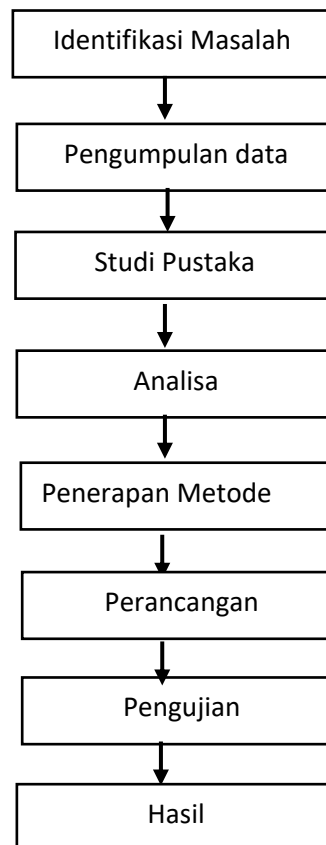
Length Encoding. lebih banyak dimanfaatkan untuk gambar. Tujuan lainnya adalah untuk mengetahui hasil kompresi apakah hasil kompresi lebih sedikit dibandingkan citra yang tidak dikompresi. karna tujuan utama adalah untuk meminimalkan *storage*[5][6].

Dalam Jurnal Teknik vol 3, No. 2, tahun 2012 yang berjudul “Implementasi Metode Run Length Encoding (RLE) untuk Kompresi Citra”[7][8]. Metode *Run Length Encoding* adalah salah satu algoritma kompresi untuk citra digital. metode kompresi *Run Length Encoding* menggunakan prinsip bahwa nilai derajat keabuan yang sering muncul di dalam *citra* akan dikodekan dengan jumlah bit yang lebih sedikit sedangkan nilai keabuan yang *frekuensi* kemunculannya sedikit akan dikodekan dengan jumlah bit yang lebih panjang. *Run Length Encoding* cocok untuk mengkompresi citra yang memiliki kelompok *pixel* berderajat keabuan yang sama[9].

2. METODOLOGI PENELITIAN

2.1 Tahapan –Tahapan Penelitian

Adapun tahapan – tahapan penelitian adalah sebagai berikut :



2.2 Kompresi

Proses kompresi merupakan proses mereduksi ukuran suatu data untuk menghasilkan representasi digital yang padat atau mampat (*compact*) namun tetap dapat mewakili kuantitas informasi yang terkandung pada data tersebut. Pada citra, video, dan audio, kompresi mengarah pada minimisasi jumlah *bit rate* untuk representasi digital. Pada beberapa literatur, istilah kompresi sering disebut juga *source coding*, *data compression*, *bandwidth compression*, dan *signal*.

2.3 Pengolahan Citra

Pengolahan citra adalah istilah umum untuk berbagai teknik yang keberadaanya untuk memanipulasi dan modifikasi citra dengan berbagai cara. foto adalah contoh gambar berdimensi dua yang dapat diolah dengan mudah. Setiap foto dalam bentuk citra digital (misalnya berasal dari kamera digital) dapat diolah melalui perangkat lunak tertentu. Sebagai contoh, apabila hasil bidikan kamera terlihat agak gelap, citra dapat diolah menjadi lebih terang.

2.4 Metode Run Length Encoding (RLE)

Algoritma RLE menggunakan pendekatan ruang. Algoritma ini cocok digunakan untuk memampatkan citra yang memiliki kelompok-kelompok piksel berderajat keabuan yang sama. Metode ini dilakukan dengan menyatakan seluruh baris citra menjadi sebuah baris run, lalu menghitung run-length untuk setiap derajat keabuan yang berurutan.

2.5. Algoritma aritmetic Coding

Algoritma aritmetic Coding melakukan penggantian suatu deretan symbol input dengan sebuah bilangan *floating point*. Semakin panjang dan semakin kompleks pesan yang dikodekan, akan semakin banyak bit yang diperlukan untuk keperluan tersebut.

3. HASIL DAN PEMBAHASAN

Pada penelitian ini, di dapat beberapa hasil dan pembahasan Untuk melakukan kompresi citra menggunakan metode Run Length Encoding dan Aritmetic terdapat beberapa proses yang akan dilakukan penulis mengimplementasikan kompresi citra dengan ukuran 3 x 3, dengan tujuan untuk mengetahui cara kerja kompresi Run Length Encoding dalam kompresi citra, maka sebagai berikut nilai decimal dari citra grayscale 3 x 3 :

Tabel 1 Tabel Citra Grayscale 3 x 3

238	255	255
238	221	221
255	204	204

Tabel 2 Tabel Citra Grayscale 3 x 3

238	255	255
238	221	221
255	204	204

Gambar 2 fahmi_grayscale.bmp

Selanjutnya mengkompresi citra grayscale diatas menggunakan metode Run Length Encoding, maka seperti inilah hasil dari kompresi citra grayscale :

(238,1) (255,2)

(238,1) (221,2)

(255,1) (204,2)

Setelah selesai melakukan kompresi menggunakan metode Run Length Encoding, selanjutnya penulis melakukan pengkodean dari data tersebut, maka seperti dibawah inilah hasil :

238,1 255,2 238,1 221,2 255,1 204, 2

Semuanya =12 pixel

Ukuran citra sebelum dikompresi 3 x 3 x 8 bit = 72 bit

Ukuran citra setelah dikompresi = 12 x 8 bit = 96 bit

Rasio pemampatan = $\left(100\% - \frac{96}{72} \times 100\%\right) = -33\%$ artinya -33% dari citra semula telah dimampatkan.

Untuk mengetahui cara kerja kompresi Aritmetic Coding, penulis mengimplementasikan dengan menggunakan data hasil kompresi Run Length Encoding, maka seperti dibawah inilah data yang akan di kompresi :

238,1 255,2 238,1 221,2 255,1 204, 2

selanjutnya penulis menentukan nilai probabilitas tiap-tiap simbol yang akan di encode menggunakan algoritma Aritmetic Coding, maka seperti inilah nilai simbol-simbol probabilitas pada table dibawah ini:

Tabel 3. Tabel Probabilitas Citra Grayscale 3 x 3

Karakter	Frekuensi	Probabilitas
1	3	$3/12 = 0,2$
2	3	$3/12 = 0,2$
204	1	$1/12 = 0,1$
221	1	$1/12 = 0,1$
238	2	$2/12 = 0,2$
255	2	$2/12 = 0,2$

Setelah probabilitas tiap karakter diketahui, tiap simbol/karakter akan diberikan range tertentu yang nilainya berkisar antara 0 dan 1, sesuai probabilitas yang ada. Dalam hal ini tidak ada ketentuan urutan penentuan segmen, asalkan antara encoder dan decoder melakukan hal yang sama. Selanjutnya, diperoleh table berikut :

Tabel 3 Tabel Probabilitas dan Range Citra Grayscale

Karakter	Frekuensi	Probabilitas	Range
1	3	$3/12 = 0,2$	$0,0 \leq 1 < 0,2$
2	3	$3/12 = 0,2$	$0,2 \leq 2 < 0,4$
204	1	$1/12 = 0,1$	$0,4 \leq 204 < 0,5$
221	1	$1/12 = 0,1$	$0,5 \leq 221 < 0,6$
238	2	$2/12 = 0,2$	$0,6 \leq 238 < 0,8$
255	2	$2/12 = 0,2$	$0,8 \leq 255 < 1,0$

Satu hal yang perlu dicatat dari table ini adalah tiap karakter melingkupi range yang disebutkan, kecuali bilangan tinggi. Sebelum penulis melakukan kompresi, terlebih dahulu penulis menerangkan maksud dari tiap karakter.

Untuk string “238,1 255,2 238,1 221,2 255,1 204, 2 ” diatas, pertama kita ambil karakter pertama yaitu “ 238”.

Low = 0,0 (kondisi awal)

High = 1,0 (kondisi awal)

$CR1 = high - low$

$= 1,0 - 0,0$

$= 1$

High_range (238) = 0,8

Low_range (238) = 0,6

Kemudian, didapatkan nilai-nilai berikut.

High = *low* + *CR*high_range* (238)

$= 0,0 + 1 * 0,8$

$= 0,8$

Low = *low* + *CR*low_range* (238)

$= 0,00 + 1 * 0,6$

$= 0,6$

Kemudian, karakter kedua “1” diambil dan diperoleh :

low = 0,6 (hasil perhitungan dari karakter 238)

high = 0,8 (hasil perhitungan dari karakter 238)

$CR2 = high - low$

$= 0,8 - 0,6$

$= 0,2$

$$High_range (1) = 0,2$$

$$Low_range (1) = 0,0$$

Kemudian, didapatkan nilai-nilai berikut.

$$High = low + CR * high_range (1)$$

$$= 0,6 + 0,2 * 0,2$$

$$= 0,64$$

$$Low = low + CR * low_range (1)$$

$$= 0,6 + 0,2 * 0,0$$

$$= 0,6$$

Kemudian, karakter ketiga “255” diambil sehingga diperoleh :

$$low = 0,6$$

$$high = 0,64$$

$$CR3 = high - low$$

$$= 0,64 - 0,6$$

$$= 0,04$$

$$High_range (255) = 1,0$$

$$Low_range (255) = 0,8$$

Kemudian, nilai-nilai berikut didapatkan.

$$High = low + CR * high_range (255)$$

$$= 0,6 + 0,04 * 1,0$$

$$= 0,64$$

$$Low = low + CR * low_range (255)$$

$$= 0,6 + 0,04 * 0,8$$

$$= 0,632$$

Kemudian, karakter keempat “2” diambil sehingga diperoleh :

$$low = 0,632$$

$$high = 0,64$$

$$CR4 = high - low$$

$$= 0,64 - 0,632$$

$$= 0,008$$

$$High_range (2) = 0,4$$

$$Low_range (2) = 0,2$$

Kemudian, nilai-nilai berikut didapatkan.

$$High = low + CR * high_range (2)$$

$$= 0,632 + 0,008 * 0,4$$

$$= 0,6352$$

$$Low = low + CR * low_range (2)$$

$$= 0,632 + 0,008 * 0,2$$

$$= 0,6336$$

Lanjutkan Sampai

Kemudian, nilai-nilai berikut didapatkan.

$$High = low + CR * high_range (2)$$

$$= 0,63459432448 + 0,00000000512 * 0,4$$

$$= 0,634594326528$$

$$Low = low + CR * low_range (2)$$

$$= 0,63459432448 + 0,00000000512 * 0,2$$

$$= 0,634594325504$$

Jika telah selesai melakukan perhitungan, maka data tersebut ditulis dalam bentuk table, data tersebut dapat terlihat sebagai berikut :

Tabel 5 Tabel Hasil Kompresi Citra Grayscale

No	Huruf	Low	High	CR
----	-------	-----	------	----

Awal		0,00	0,94	0,94
1	238	0,6	0,8	0,1
2	1	0,6	0,64	0,2
3	225	0,632	0,64	0,04
4	2	0,6336	0,6352	0,008
5	238	0,63456	0,63488	0,0016
6	1	0,63456	0,634624	0,00032
7	221	0,634592	0,6345984	0,000064
8	2	0,63459328	0,63459456	0,0000064
9	255	0,634594304	0,63459456	0,00000128
10	1	0,634594304	0,6345943552	0,000000256
11	204	0,63459432448	0,6345943296	0,0000000512
12	2	0,634594325504	0,634594326528	0,00000000512

Dalam hal ini simbol habis bisa ditandai dengan simbol khusus, misalnya *End of Message* (ES) atau dengan menyatakan panjang pesan.

Untuk pesan yang telah diencode, proses decoding berikut dilakukan.

ES= 0,634594325504 Bandingkan nilai ini dengan simbol berikut :

Tabel 6 Tabel Probabilitas dan Range Citra Grayscale

Karakter	Frekuensi	Probabilitas	Range
1	3	$3/12 = 0,2$	$0,0 \leq 1 < 0,2$
2	3	$3/12 = 0,2$	$0,2 \leq 2 < 0,4$
204	1	$1/12 = 0,1$	$0,4 \leq 204 < 0,5$
221	1	$1/12 = 0,1$	$0,5 \leq 221 < 0,6$
238	2	$2/12 = 0,2$	$0,6 \leq 238 < 0,8$
255	2	$2/12 = 0,2$	$0,8 \leq 255 < 1,0$

Nilai ES = 0,634594325504, didapatkan range simbol yang pertama, yaitu simbol/karakter “238”.

$$Low_range = 0,6$$

$$High_range = 0,8$$

$$CR = high_range - low_range$$

$$= 0,8 - 0,6$$

$$= 0,2$$

$$ES = ES - low_range$$

$$= 0,634594325504 - 0,6$$

$$= 0,034594325504$$

$$ES = ES/CR$$

$$ES = 0,034594325504 / 0,2$$

$$ES = 0,17297162752$$

Nilai ES = 0,17297162752, didapatkan range simbol yang kedua, yaitu simbol/karakter “1”.

$$Low_range = 0,0$$

$$High_range = 0,2$$

$$CR = high_range - low_range$$

$$= 0,2 - 0,0$$

$$= 0,2$$

$$ES = ES - low_range$$

$$= 0,17297162752 - 0,0$$

$$= 0,17297162752$$

$$ES = ES / CR$$

$$ES = 0,17297162752 / 0,2$$

$$ES = 0,8648581376$$

Nilai ES = 0,8648581376, didapatkan range simbol yang ketiga, yaitu simbol/karakter “255”.

$$Low_range = 0,8$$

$$High_range = 1,0$$

$$\begin{aligned} CR &= high_range - low_range \\ &= 1,0 - 0,8 \\ &= 0,2 \end{aligned}$$

$$\begin{aligned} ES &= ES - low_range \\ &= 0,8648581376 - 0,8 \\ &= 0,0648581376 \end{aligned}$$

$$\begin{aligned} ES &= ES / CR \\ ES &= 0,0648581376 / 0,2 \\ ES &= 0,324290688 \end{aligned}$$

Lanjutkan Sampai

Nilai ES = 0,084, didapatkan range simbol yang kesepuluh, yaitu simbol/karakter “1”.

$$\begin{aligned} Low_range &= 0,0 \\ High_range &= 0,2 \end{aligned}$$

$$\begin{aligned} CR &= high_range - low_range \\ &= 0,2 - 0,0 \\ &= 0,2 \end{aligned}$$

$$\begin{aligned} ES &= ES - low_range \\ &= 0,084 - 0,0 \\ &= 0,084 \end{aligned}$$

$$\begin{aligned} ES &= ES / CR \\ ES &= 0,084 / 0,2 \\ ES &= 0,42 \end{aligned}$$

Nilai ES = 0,42, didapatkan range simbol yang kesebelas, yaitu simbol/karakter “204”.

$$\begin{aligned} Low_range &= 0,4 \\ High_range &= 0,5 \end{aligned}$$

$$\begin{aligned} CR &= high_range - low_range \\ &= 0,5 - 0,4 \\ &= 0,1 \end{aligned}$$

$$\begin{aligned} ES &= ES - low_range \\ &= 0,42 - 0,4 \\ &= 0,02 \end{aligned}$$

$$\begin{aligned} ES &= ES / CR \\ ES &= 0,02 / 0,1 \\ ES &= 0,2 \end{aligned}$$

Nilai ES = 0,2, didapatkan range simbol yang keduabelas, yaitu simbol/karakter “2”.

$$\begin{aligned} Low_range &= 0,2 \\ High_range &= 0,4 \end{aligned}$$

$$\begin{aligned} CR &= high_range - low_range \\ &= 0,4 - 0,2 \\ &= 0,2 \end{aligned}$$

$$\begin{aligned} ES &= ES - low_range \\ &= 0,2 - 0,2 \\ &= 0 \end{aligned}$$

$$\begin{aligned} ES &= ES / CR \\ ES &= 0 / 0,2 \\ ES &= 0 \end{aligned}$$

Sampai disini perhitungan dihentikan karena diperoleh nilai ES=0. Bila dibuat dengan bentuk table maka diperoleh hasil sebagai berikut :

Tabel 7 Tabel Hasil Dekompresi Citra Grayscale

No	ES	Huruf	Low	High	CR
1	0,634594325504	238	0,6	0,8	0,2

2	0,17297162752	1	0,0	0,2	0,2
3	0,8648581376	225	0,8	1,0	0,2
4	0,324290688	2	0,2	0,4	0,2
5	0,62145344	238	0,6	0,8	0,2
6	0,1072672	1	0,0	0,2	0,2
7	0,536336	221	0,5	0,6	0,1

Tabel 8 Tabel Hasil Dekompresi Citra Grayscale

No	ES	Huruf	Low	High	CR
8	0,36336	2	0,2	0,4	0,2
9	0,8168	255	0,8	1,0	0,2
10	0,084	1	0,0	0,2	0,2
11	0,42	204	0,4	0,5	0,1
12	0,2	2	0,2	0,4	0,2
13	0 (selesai)	-	-	-	-

Terlebih dahulu penulis mengambil data hasil citra yang sudah dekompresi menggunakan Aritmetic Coding, maka seperti dibawah inilah hasil kompresi Aritmetic Coding :

Low = 0,634594325504

Selanjutnya penulis merubah Low ke bentuk string awal yaitu :

238,1 255,2 238,1 221,2 255,1 204, 2

Selanjutnya penulis melakukan dekompresi menggunakan metode Run Length Encoding, maka seperti dibawah inilah hasilnya :

(238,1) (255, 2)

(238,1) (221,2)

(255,1) (204,2)

Setelah selesai melakukan dekompresi, selanjutnya penulis mengembalikan ke table awal, seperti dibawah ini :

Tabel 3.9 Tabel Citra Grayscale 3 x 3

238	255	255
238	221	221
255	204	204

jika tidak berurutan nilai baris citra, maka hasil kompresinya menjadi lebih banyak dari pada citra yang tidak di kompresi, maka contoh naya seperti dibawah ini :

Tabel 3.10 Tabel Citra Grayscale 3 x 3

238	255	255
238	221	221
255	204	204

Dari table diatas, maka dilakukan proses kompres menggunakan metode Run Length Encoding maka seperti diwah ini hasil nya :

238,1 255,2 238,1 221,2 255,1 204, 2

Semuanya =12 pixel

Ukuran citra sebelum dikompresi 3 x 3 x 8 bit = 72 bit

Ukuran citra setelah dikompresi = 12 x 8 bit = 96 bit

Rasio pemampatan = $\left(100\% - \frac{96}{72} \times 100\%\right) = -33\%$ artinya -33% dari citra semula telah dimampatkan.

Keterangan:

1. Metode Run Length Encoding cocok untuk kompresi citra yg memiliki deretan nilai pixel yang sama dalam satu baris.
2. Hasil kompresi metode Run Length Encoding akan menjadi lebih banyak jika deretan nilai pixel berbeda-beda dalam satu baris.

Contoh :

		
Original size: 10000 bytes Compressed size: 5713 bytes Ratio: 1.75	Original size: 10000 bytes Compressed size: 10100 Ratio: 0.99	Original size: 10000 bytes Compressed size: 200 Ratio: 50

Gambar 5 Gambar Perbandingan Hasil Kompresi

Untuk mengetahui tingkat rasio kompresi menggunakan Aritmetik Coding, maka diambil nilai dari hasil kompresi Aritmetic Coding seperti dibawah ini:

0,634594325504

Semuanya =13 pixel

Ukuran citra sebelum dikompresi $12 \times 8 \text{ bit} = 96 \text{ bit}$

Ukuran citra setelah dikompresi $= 13 \times 8 \text{ bit} = 104 \text{ bit}$

Rasio pemampatan $= \left(100\% - \frac{104}{96} \times 100\%\right) = -8\%$ artinya -8% dari citra semula telah dimampatkan.

4. KESIMPULAN

Berdasarkan kesimpulan yang dapat diambil setelah dilakukan pembahasan maka diambil kesimpulan dari metode Run Length Encoding dalam mengkompresi citra yaitu dengan membaca nilai pixel berdasarkan baris dan menambah bit penanda di setiap pixel, untuk menentukan bit penanda di setiap pixel itu tergantung nilai pixelnya sendiri, semangkin banyak nilai pixel yang sama maka semangkin besarlah bit penanda ditulis. Cara kerja algoritma Aritmetic coding dalam mengkompresi citra dengan, semangkin banyak nilai pixel yang sama dalam satu citra dan saling berdekatan, maka semangkin efisien lah mengkompresi citra menggunakan metode Run Length Encoding, tapi sebaliknya, hasil kompresi akan semangkin jelek jika objek citranya berbeda-beda warna dalam setiap pixelnya. Tingkat efisiensi dari algoritma Aritmetic Coding dalam kompresi citra itu dilihat dari hasil *plotting point* nya, semangkit panjang nilai yang dikompresi maka semangkin panjang pula *plotting point* yang didapat. Dalam

REFERENCES

- [1] P. Rianto dan A. Harjoko, "Penentuan Kematangan Buah Salak Pondoh Di Pohon Berbasis Pengolahan Citra Digital," *IJCCS (Indonesian J. Comput. Cybern. Syst.*, vol. 11, no. 2, hal. 143, 2017, doi: 10.22146/ijccs.17416.
- [2] M. Irfan, B. A. Ardi Sumbodo, dan I. Candradewi, "Sistem Klasifikasi Kendaraan Berbasis Pengolahan Citra Digital dengan Metode Multilayer Perceptron," *IJEIS (Indonesian J. Electron. Instrum. Syst.*, vol. 7, no. 2, hal. 139, 2017, doi: 10.22146/ijeis.18260.
- [3] Yelly N Nabuasa, "Pengolahan Citra Digital Perbandingan Metode Histogram Equalization Dan Spesification Pada Citra Abu-Abu," *J-Icon*, vol. 7, no. 1, hal. 87–95, 2019.
- [4] M. Simangunsong, "Perbandingan Algoritma Elias Delta Code Dan Unary Coding Dalam Kompresi Citra Forensik," vol. 12, no. 1, hal. 18–26, 2020.
- [5] J. Jumadi, Y. Yupianti, dan D. Sartika, "Pengolahan Citra Digital Untuk Identifikasi Objek Menggunakan Metode Hierarchical Agglomerative Clustering," *JST (Jurnal Sains dan Teknol.*, vol. 10, no. 2, hal. 148–156, 2021, doi: 10.23887/jstundiksha.v10i2.33636.
- [6] L. V. Simanjuntak, "Perbandingan Algoritma Elias Delta Code dengan Levenstein Untuk Kompresi File Teks," vol. 1, no. 3, hal. 184–190, 2020.
- [7] W. Gazali, H. Soeparno, dan J. Ohliati, "Dalam Pengolahan Citra Digital," *J. Mat Stat*, vol. 12, no. 2, hal. 103–113, 2014.
- [8] S. Ratna, "Pengolahan Citra Digital Dan Histogram Dengan Phyton Dan Text Editor Phycharm," *Technol. J. Ilm.*, vol. 11, no. 3, hal. 181, 2020, doi: 10.31602/tji.v11i3.3294.
- [9] O. N. Shpakov dan G. V. Bogomolov, "Technogenic activity of man and local sources of environmental pollution," *Stud. Environ. Sci.*, vol. 17, no. C, hal. 329–332, 1981, doi: 10.1016/S0166-1116(08)71924-1.