TIN: Terapan Informatika Nusantara
Vol 2, No 9, Februari 2022, Hal 527−532
ISSN 2722-7987 (Media Online)
Website https://ejurnal.seminar-id.com/index.php/tin
DOI 10.47065/tin.v2i9.1027

Implementasi Algoritma Arithmetic Coding dan Sannon-Fano Pada Kompresi Citra PNG

Ibnu Syuhada

Program Studi Manajemen Informatika Universitas Budi Darma Medan, Indonesia
Email: Ibnusyuhada21@gmail.com

Abstrak−Perkembangan teknologi yang pesat, sangat berperan penting dalam pertukaran informasi yang cepat. Pada pengiriman informasi dalam bentuk citra masih mengalami kendala, diantaranya adalah karena besarnya ukuran citra sehingga solusi untuk masalah tersebut adalah dengan melakukan kompresi. Pada skripsi ini akan mengimplementasi dan membandingkan kinerja algoritma *Arithmetic Coding* dan *Shannon-Fano*melalui perhitungan rasio kompresi, ukuran*file* hasil kompresi, kecepatan proses kompresi dan dekompresi. Berdasarkan seluruh hasil pengujian, bahwa algoritma *Arithmetic Coding* menghasilkan rata-rata rasio kompresi 62,88 %dan rasio kompresi *Shannon-Fano*61,73 %, kemudian *Arithmetic Coding* rata-rata kecepatan dalam kompresi citra yaitu 0,072449 detik dan *Shannon-Fano*0,077838 detik. Kemudian algoritma *Shannon-Fano* memiliki rata-rata kecepatan untuk dekompresi yaitu 0,028946 detik dan algoritma *Arithmetic Coding* 0,034169 detik. Citra hasil dekompresi pada algoritma *Arithmetic Coding* dan *Shannon-Fano* sesuai dengan citra asli.Dapat diambil kesimpulan dari hasil pengujian bahwa algoritma *Arithmetic Coding* lebih efisien dalam mengkompresi citra *.png dibandingkan algoritma *Shannon-Fano,* walaupun dalam hal dekompresi *Shannon-Fano*se dikit lebih cepat dibandingkan *Arithmetic Coding*.

Kata Kunci: Kompresi Citra; Arithmetic Coding; Shannon-Fano

Abstract− The rapid development of technology plays an important role in the rapid exchange of information. In sending information in the form of images, there are still problems, including because of the large size of the image so that the solution to this problem is to perform compression. In this thesis, we will implement and compare the performance of the Arithmetic Coding and Shannon-Fano algorithms by calculating the compression ratio, compressed file size, compression and decompression process speed. Based on all test results, that the Arithmetic Coding algorithm produces an average compression ratio of 62.88% and a Shannon-Fano compression ratio of 61.73%, then Arithmetic Coding the average speed in image compression is 0.072449 seconds and Shannon-Fano 0.077838 second. Then the Shannon-Fano algorithm has an average speed for decompression of 0.028946 seconds and the Arithmetic Coding algorithm 0.034169 seconds. The decompressed image on the Arithmetic Coding and Shannon-Fano algorithm is in accordance with the original image. It can be concluded from the test results that the Arithmetic Coding algorithm is more efficient in compressing *.png images than the Shannon-Fano algorithm, although in terms of decompression Shannon-Fanose is a little faster compared to Arithmetic Coding.

Keywords: Image Compression; Arithmetic Coding; Shannon-Fano

1. PENDAHULUAN

Pada umumnya citra png terdiri dari 4 blok data yaitu: PNG header, Bit Information (DIB header), Color Pallete, dan Png Data. PNG header berisi informasi umum dari citra png. Blok ini berada pada bagian awal file citra dan digunakan untuk mengidentifikasi citra. Beberapa aplikasi pengolah citra akan membaca blok ini untuk memastikan bahwa citra tersebut berformat png dan tidak dalam kondisi rusak. Bit information berisi informasi detail dari citra png, yang akan digunakan untuk menampilkan citra pada layar [1].

Kompresi adalah proses pengubahan sekumpulan data menjadi suatu bentuk kode atau symbol untuk menghemat tempat penyimpanan dan waktu. Algoritma kompresi yang diharapkan dari kompresi citra adalah proses kompresi dan dekompresinya cepat, meminimalkan pemakaian memory, kualitas citra yang baik dan proses transfer yang mudah.

Pada umumnya representasi citra membutuhkan memori yang besar, semakin besar ukuran citra tentu semakin besar memori yang dibutuhkan. Pada sisi lain,kebanyakan citra terdapat duplikasi data. Duplikasi data pada citra yaitu ada 2 hal Pertama, besar kemungkinan suatu piksel dengan piksel tetangganya memiliki intensitas yang sama, sehingga penyimpanan piksel memboroskan tempat. Kedua,citra banyak mengandung bagian yang sama, sehingga bagian yang sama ini tidak dikodekan berulang kali.

Alasan penulis mengunakan kedua metode ini karena kedua metode ini memliki kelebihannya masing-masing. Kedua metode yang digunakan penulis diharapkan akan memiliki kelebihan yaitu sebagai teknik kompresi data tanpa menghilangkan satu pun informasi saat sebelum dikompresi. Pengiriman informasi dalam bentuk citra masih mengalami kendala dalam hal ukuran yaitu citra yang memiliki ukuran yang besar sehingga sulit untuk melakukan pengiriman dan pemindahan oleh karena itu penelitiakan melakukan penelitian tentang bagaimana mengkompresi ukuran citra, agar mempermudah pengiriman dan pemindahan pada perangkat keras yang ada.

Berdasarkan masalah yang telah diuraikan di atas, peneliti akan menggunakan algoritma Arithmetic Coding dengan Shannon-Fano untuk mengkompresi citra karena menurut penelitian sebelumnya dengan menggunakan kedua algoritma tersebut diharapkan akan memiliki kelebihan yaitu mengkompresi citra menjadi ukuran yang lebih kecil dan citra yang dihasilkan setelah dekompresi memiliki kualitas yang sama dengan citra sebelum dikompresi, setelah itu peneliti akan membandingkan dan menganalisis kinerja kedua algoritma dalam hal rasio kompresi serta lama waktu kompresi dan dekompresi berektensi .* png.

Dari hasil penelitian yang sebelumnya juga menggunakan metode Arithmetic Coding antara lain seperti Studi Kompresi Data dengan Metode Arithmetic Coding. Hasil penelitian menunjukkan bahwa algoritma ini cukup baik untuk dipakai dalam keperluan kompresi data.Alasan pertama karena jumlah coding bit pada arithmatic coding lebih sedikit

TIN: Terapan Informatika Nusantara
Vol 2, No 9, Februari 2022, Hal 527−532
ISSN 2722-7987 (Media Online)
Website https://ejurnal.seminar-id.com/index.php/tin
DOI 10.47065/tin.v2i9.1027

dibandingkan dengan Huffman Coding. Modifikasi dengan menggunakan bilangan integer juga mampu mengatasi keterbatasan peralatan-peralatan encoder dan decoder dari pengolahan floating point yang terlalu panjang. Kedua karena jumlah bit kodenya lebih sedikit dan dapat diimplementasikan [2]. Dan Analisis Perbandingan Kompresi dan Dekompresi Menggunakan Algoritma Shannon-Fano 2 Gram Dan Lempel Ziv Welch Pada Terjemahan Hadits Shahih Muslim. Hasil penelitian menunjukkan bahwa secara rata-rata, algoritma Lempel Ziv Welch menghasilkan rasio file yang lebih baik sekitar± 45,72 % dibandingkan dengan Algoritma Shannon-Fano 2 Gram yang hanya menghasilkan ± 58,50 % [3].

2. METODOLOGI PENELITIAN

2.1 Citra

Citra adalah suatu representasi (gambaran), kemiripan atau imitasi dari suatu objek. Citra sebagai keluaran suatu system perekaman data dapat bersifat optic berupa foto, bersifat analog berupa sinyal-sinyal video seperti gambar pada monitor televise atau bersifat digital yang dapat langsung disimpan pada suatu media penyimpan[5]. Citra dapat dikelompokkan menjadi dua macam, yaitu citra analog dan citra digital. Citra analog dihasilkan dari sistem optic yang menerima sinyal analog. Contoh: mata manusia, kamera analog. Sedangkan citra digital dihasilkan melalui proses digitalisasi terhadap citra kontinu. Contoh: kamera digital, *scanner*.

2.2 Kompresi

Kompresi data adalah sebuah cara untuk memadatkan data sehingga hanya memerlukanruangan penyimpanan lebih kecil sehingga lebih efisien dalam menyimpannya ataumempersingkat waktu pertukaran data tersebut. Proses kompresi didasarkan padakenyataan bahwa hampir semua jenis data selalu terdapat pengulangan pada komponendata yang dimilikinya, misalnya di dalam suatu citra akan terdapat pengulangan warna dari 0 hingga 255. Melalui proses kompresi berusaha untuk menghilangkan unsur pengulangan ini dengan mengubahnya sedemikian rupa sehingga ukuran data menjadi lebih kecil.

2.3 Citra *PNG*

Format PNG, disebut dengan png atau format DIB (*Device Independent portable network graphics*) adalah sebuah format citra yang digunakan untuk menyimpan citra png digital terutama pada sistem operasi Microsoft Windows atau OS/2. Pada citra berformat *.png (*portable network graphics*) yang tidak terkompresi, piksel citra disimpan dengan kedalaman warna 1, 4, 8, 16 atau 24 bit per piksel [10]. Model ruang warna yang digunakan pada citra png adalah RGB (*red, green, dan blue*). Setiap komponen panjangnya 8 bit, jadi ada 256 nilai keabuan untuk warna merah, 256 nilai keabuan untuk warna hijau, 256 nilai keabuan untuk warna biru. Nilai setiap piksel tidak menyatakan derajat keabuan secara langsung, tetapi nilai piksel menyatakan indeks tabel RGB yang memuat nilai keabuan merah (R), nilai keabuan hijau (G), nilai keabuan biru (B) untuk masing-masing piksel yang bersangkutan. Namun pada citra hitam-putih, nilai R = G = B untuk menyatakan bahwa citra hitam putih hanya mempunya satu kanal warna. Citra hitam putih umumnya adalah citra 8 bit. Citra yang lebih kaya warna adalah citra 24 bit. Setiap piksel panjangnya 24 bit, karena setiap piksel langsung menyatakan komponen warna merah, komponen warna hijau, dan komponen warna biru. Masing-masing komponen panjangnya 8 bit. Citra 24 bit disebut juga citra 16 juta warna, karena citra ini mampu menghasilkan 224 = 16.777.216 kombinasi warna [1].

2.4 Algoritma *Encoding*

Algoritma untuk menyelesaikan penghitungan output angka dari *message* tersebut adalah:
1. Set *Low number* pada angka 0
2. Set *High number* pada angka 1
3. Lakukan perulangan hingga semua simbol terproses

Range = High – Low
High = Low + Range * Angka *range* tertinggi dari simbol yang sedang diproses
Low = Low + Range * Angka *range* terendah dari simbol yang sedang diproses
Dari hasil *encoding* keseluruhan untuk *message* “tawa” didapatkan:

Tabel 1. Hasil *Encoding* Untuk Message “Tawa”

Simbol	*Range*	*Low Number* 0	*High Number* 1
t	1	0.5	0.75
a	0.25	0.5	0.625
w	0.125	0.59375	0.625
a	0.3125	0.59375	0.609375

Jadi angka terakhir dari *low number* yaitu 0.59375 merupakan hasil output angka proses *encoding* dari *message* “tawa”.

Setelah mengetahui pola *encoding* tersebut, maka dengan mudah dapat dilihat bagaimana proses *decoding* bekerja. Dalam proses *decoding* ini pertama-tama temukan simbol pertama yaitu dengan melihat output angka dari proses

TIN: Terapan Informatika Nusantara
Vol 2, No 9, Februari 2022, Hal 527−532
ISSN 2722-7987 (Media Online)
Website https://ejurnal.seminar-id.com/index.php/tin
DOI 10.47065/tin.v2i9.1027

encoding itu jatuh pada *range* simbol mana. Karena output angka 0.59375 jatuh pada range 0.5-0.75, dapat diketahui bahwa simbol pertama adalah "t". Kemudian yang harus dilakukan yaitu mengeluarkan "t" yang telah ter-*encode* dengan cara membalikkan proses yang menempatkannya. Pertama, dengan mengurangi *high number* dan *low number* yang dimiliki simbol "t" untuk mendapatkan angka *range*nya dan angka yang didapat yaitu 0.25. Selanjutnya mengurangi output angka *encoding* dengan angka *range* terendah simbol "t" dan didapatkan angka 0.9375 untuk kemudian membagi angka tersebut dengan *range* yang sebelumnya pernah didapat. Hasil yang didapat adalah 0.375. Dan langkah selanjutnya dengan memperhitungkan angka tersebut jatuh pada *range* symbol mana, dan didapatkan simbol berikutnya yaitu "a".

2.5 Algoritma *Decoding*

Algoritma untuk proses *decoding* yaitu :
1. Dapatkan angka *encoded* (pada variabel "*Numbe*r")
2. Dapatkan simbol dimana angka *encoded* ("*Number*") berada dalam *range*-nya
3. Lakukan perulangan hingga semua simbol terproses.

Range = angka *range* tertinggi simbol – angka *range* terendah simbol
Number = *Number* - angka *range* terendah simbol
Number = *Number* / *Range*
Dari hasil proses *decoding* untuk *message* "tawa" dapat dilihat pada tabel berikut :

Tabel 2. Hasil *Decoding* Untuk Message "Tawa"

Simbol	*Low Range*	*High Range*	*Range*	*Number*
t	0.5	0.75	0.25	0.59375
a	0	0.5	0.5	0.375
w	0.75	1	0.25	0.75
a	0	0.5	0.5	0

Ketika tidak ada lagi simbol yang tersisa untuk di-*decode*, untuk menghentikannya yaitu dapat dengan menyediakan *range* kecil khusus untuk simbol EOF (*End of File*). Tetapi dalam hal pengukuran *encoding* tidak dibutuhkan dan juga sebagian besar kompresor tahu kapan harus berhenti.

Proses *encoding* pada metode *arithmetic* ini merupakan cara penyempitan *range* angka suatu simbol dengan simbol yang baru. *Range* baru tersebut sebanding dengan probabilitas yang diberikan pada simbol tersebut. Proses *decoding* adalah prosedur kebalikannya, dimana *range* dikembangkan dalam perbandingan terhadap probabilitas dari setiap simbol ketika diekstrak [7].

2.6 Algoritma *Shannon-Fano*

Algoritma *Shannon-Fano* dinamai berdasarkan nama pengembangnya yaitu Claude Shannon dan Robert Fano. Metode ini dimulai dengan deretan dari simbol n dengan kemunculan frekuensi yang diketahui. Mula-mula simbol disusun secara menaik (*ascending order*) berdasarkan frekuensi kemunculannya. Lalu set simbol tersebut dibagi menjadi dua bagian yang berbobot sama atau hampir sama. Seluruh simbol yang berada pada *subset* I diberi biner 0, sedangkan simbol yang berada pada *subset* II diberi biner 1. Setiap *subset* dibagi lagi menjadi dua *subsubset* dengan bobot kemunculan frekuensi yang kira-kira sama, dan biner kedua diberikan seperti *subset* I dan *subset* II. Ketika *subset* hanya berisi dua simbol, biner diberikan pada setiap simbol. Proses akan berlanjut sampai tidak ada *subset* yang tersisa [7]. Algoritma *Shannon-Fano* merupakan kompresi yang bersifat *lossless*, dimana metode ini harus mendekompresi berkas agar dapat direkonstruksikan menjadi berkas semula tanpa kehilangan informasi.

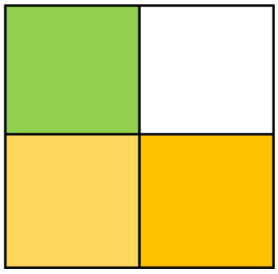
3. HASIL DAN PEMBAHASAN

Masalah utama dalam penelitian ini adalah bagaimana mengimplementasikan algoritma *Arithmetic Coding* dan *Shannon-Fano* untuk kompresi citra.Pengiriman informasi dalam bentuk citra masih mengalami kendala dalam hal ukuranyaitu citra yang memiliki ukuran yang besar sehingga sulit untuk melakukanpengiriman dan pemindahan oleh karena itu peneliti akan melakukan penelitiantentang bagaimana mengkompresi ukuran citra, agar mempermudah pengiriman danpemindahan pada perangkat keras yang ada.

Proses *encoding*, algoritma *Arithmetic Coding*:
Langkah 1: Set *low* = 0.0 (kondisi awal)
Langkah 2: Set *high* = 1.0 (kondisi awal)
Langkah 3: *While* (simbol *input* masih ada) *do*
Langkah 4: Ambil simbol *input*
Langkah 5: *Code-Range* = *high* – *low*
Langkah 6: *High* = *low* + *Code-Range***high_range* (simbol)
Langkah 7: *Low* = *low* + *Code-Range***low_range* (simbol)
Langkah 8: *End While*
Langkah 9: *Output Low*

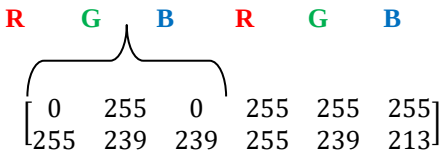
TIN: Terapan Informatika Nusantara
Vol 2, No 9, Februari 2022, Hal 527−532
ISSN 2722-7987 (Media Online)
Website https://ejurnal.seminar-id.com/index.php/tin
DOI 10.47065/tin.v2i9.1027

Proses *decoding*, algoritma *Arithmetic Coding*:
Langkah 1: Ambil *Encoded-Symbol* (ES)
Langkah 2: *Do*
Langkah 3: Cari *range* dari simbol yang melingkupi *Encoded-Symbol*(ES)
Langkah 4: Cetak simbol
Langkah 5: *Code-Range = high_range – low_range*
Langkah 6: *Encoded-Symbol = Encoded-Symbol - low_range*
Langkah 7: *Encoded-Symbol = Encoded-Symbol / Code-Range*
Langkah 8: *Until* simbol habis



Gambar 1. Citra Berwarna 2 x 2 Piksel

Diproses dalam bentuk matriks:



Gambar 2. Matriks Citra Berwarna 2 x 2 Piksel

Tabel 3. Tabel Probabilitas dan *Range* untuk Gambar 3.1

No	Simbol	Frekuensi	Probabilitas	*Range*
1	0	2	2/12=0,17	0,0≤0<0,17
2	255	6	6/12=0,5	0,17≤255<0,67
3	239	3	3/12=0,25	0,67≤239<0,92
4	213	1	1/12=0,08	0,92≤213<1,00
Total		12		

Keterangan:
0,0≤0<0,17 : Nilai “0” memiliki *range* dari 0,0 sampai dengan 0,17
0,17≤255<0,67 : Nilai “255” memiliki *range* dari 0,17 sampai dengan 0,67
0,67≤239<0,92 : Nilai “239” memiliki *range* dari 0,67 sampai dengan 0,92
0,92≤213<1,00 : Nilai “213” memiliki *range* dari 0,92 sampai dengan 1

Tabel 4. Proses *Encoding* untuk Gambar 3.8

No Awal	Simbol	*Low* 0,0	*High* 1,0	CR 1,0
1	0	0,0	0,17	1,0
2	255	0,0289	0,1139	0,17
3	0	0,0289	0,04335	0,085
4	255	0,0313565	0,0385815	0,01445
5	255	0,03258475	0,03619725	0,007225
6	255	0,033198875	0,035005125	0,0036125
7	255	0,0335059375	0,0344090625	0,00180625
8	239	0,03411103125	0,0343368125	0,000903125
9	239	0,0342623046875	0,03431875	0,00022578125
10	255	0,034271900390625	0,034300123046875	0,0000564453125
11	239	0,0342908095703125	0,034297865234375	0,00002822265625
12	213	0,03429730078125	0,034297865234375	0,0000070556640625

Tabel 5. Tabel *Range* Probabilitas

TIN: Terapan Informatika Nusantara
Vol 2, No 9, Februari 2022, Hal 527−532
ISSN 2722-7987 (Media Online)
Website https://ejurnal.seminar-id.com/index.php/tin
DOI 10.47065/tin.v2i9.1027

No	Nilai	Frekuensi	Probabilitas	*Range*
1	0	2	2/12=0,17	0,0≤0<0,17
2	255	6	6/12=0,5	0,17≤255<0,67
3	239	3	3/12=0,25	0,67≤239<0,92

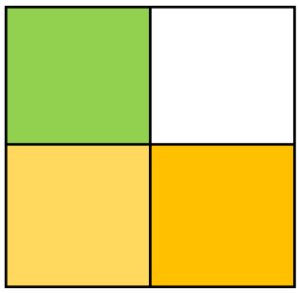
Tabel 6. Tabel *Range* Probabilitas Lanjutan

No	Nilai	Frekuensi	Probabilitas	*Range*
4	213	1	1/12=0,08	0,92≤213<1,00

Tabel 7. Proses *Decoding* untuk Gambar 3.8

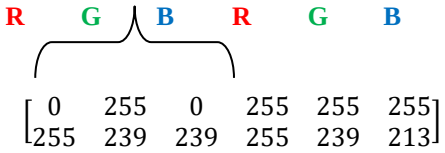
No	ES	Simbol	*Low*	*High*	CR
1	0,03429730078125	0	0,0	0,17	0,17
2	0,201748828125	255	0,17	0,67	0,5
3	0,06349765625	0	0,0	0,17	0,17
4	0,373515625	255	0,17	0,67	0,5
5	0,40703125	255	0,17	0,67	0,5
6	0,4740625	255	0,17	0,67	0,5
7	0,608125	255	0,17	0,67	0,5
8	0,87625	239	0,67	0,92	0,25
9	0,825	239	0,67	0,92	0,25
10	0,62	255	0,17	0,67	0,5
11	0,9	239	0,67	0,92	0,25
12	0,92	213	0,92	1,00	0,08
13	0,0 (Selesai)	-	-	-	-

1. Kompresi *Shannon-Fano* pada Citra
   Contoh penggunaan algoritma *Shannon-Fano* pada citra adalah misalkan pada citraberwarna dengan ukuran 2 x 2 piksel dapat dilihat pada gambar 3.9. frekuensi, asalkan antara *encoder* dan *decoder* melakukan hal yangsama.



Gambar 3. Citra Berwarna 2 x 2 Piksel

Diproses dalam bentuk matriks:



Gambar 4. Matriks Citra Berwarna 2 x 2 Piksel

Tabel 8. Frekuensi Kemunculan Simbol

Simbol	Frekuensi
255	6
239	3
0	2
213	1

Tabel 9. Proses Pengkodean

Simbol	Frekuensi	SF *Code*
255	6	0 (Pembagian Pertama)
239	3	10 (Pembagian Kedua)
0	2	110 (Pembagian Ketiga)

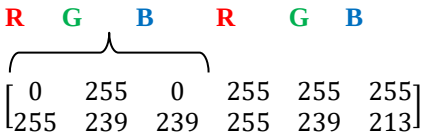
TIN: Terapan Informatika Nusantara
Vol 2, No 9, Februari 2022, Hal 527−532
ISSN 2722-7987 (Media Online)
Website https://ejurnal.seminar-id.com/index.php/tin
DOI 10.47065/tin.v2i9.1027

213	1	111
Total	12	

Tabel 10. Kode dan Panjang Kode *Shannon-Fano*

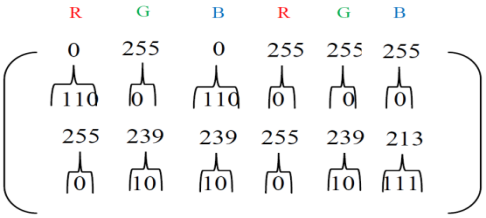
Simbol	Frekuensi	SF *Code*	Panjang *Code*
255	6	0	1 bit
239	3	10	2 bit
0	2	110	3 bit
213	1	111	3 bit
Total	12		6 bit

Selanjutnya gantikan tiap simbol pada gambar 3.8 dengan kode *Shannon-Fano*direpresentasikan dalam bentuk matriks sebagai berikut:



Gambar 5. Matriks Citra Berwarna 2 x 2 Piksel setelah di Subtitusi ke *SF Code*

2. Dekompresi *Shannon-Fano* pada Citra
   Untuk mengembalikan citra terkompresi menjadi data citra aslinya, maka diperlukanproses dekompresi. Selanjutnya untuk melakukan dekompresi yaitu dengan membacabit pertama di gambar 3.10 yaitu 1 kemudian akan dilakukan penelusuran pada *SFCode*, jika 1 tidak mewakili simbol warna maka akan baca bit berikutnya sehingga bityang dibaca menjadi 11 dilakukan penelusuran pada *SF Code*. Dan kemudian jika 11tidak mewakili simbol warna yang bersesuaian maka akan dibaca bit berikutnya sehingga bit yang dibaca menjadi 110 dan kembali dilakukan penelusuran pada *SFCode*, karena 110 mewakili simbol warna yaitu 255, sehingga 110 disubtitusikan kesimbol 255. Setelah itu dibaca kembali bit berikutnya dan lakukan penelusurankembali terhadap *SF Code* sehingga didapat bit-bit yang mewakili sebuah symbolwarna yang bersesuaian. Setelah melakukan subtitusi dari *SF Code* ke simbol warna,maka akan kembali ke citra asli. Proses ini dapat dilihat dari gambar 3.12 sebagaiberikut.



Gambar 6. Proses Subtitusi *SF Code* ke Simbol Warna

4. KESIMPULAN

Berdasarkan hasil penelitian diambil kesimpulan adanya proses kompressi citra, maka dapat dilhat hasil sebagai proses Kompresi citradengan mengkombinasikan metode kompresi. Dengan menerapkan *Algoritma Aritmetic dan Shanon-Fano*dan dapat melakukan proses kompressi citradengan baik.

REFERENCES

[1] D. R. ARDANI, "IMPLEMENTASI DAN ANALISIS ALGORITMA ZHU-TAKAOKA DAN ALGORITMA KNUTH-MORRIS-PRATT PADA APLIKASI KAMUS ISTILAH KESEHATAN BERBASIS ANDROID." Medan, p. 65, 2017.
[2] F. A. SYUHADA, *IMPLEMENTASI ALGORITMA ZHU-TAKAOKA PADA APLIKASI TERJEMAHAN AL-QURAN BERBASIS ANDROID*. Medan: USU, 2016.
[3] S. Bahri, "Konsep Implementasi Syariat Islam di Aceh," *Kanun J. Ilmu Huk.*, no. 60, pp. 313–337, 2013.
[4] Masrokhin, "FIQH SEBAGAI PRODUK IJTIHADI TERHADAP PENAFSIRAN AL-QUR'AN," pp. 1–19, 2017.
[5] Mesran, "Implementasi Algoritma Brute Force Dalam Pencarian Data Katalog Buku Perpustakaan," *Maj. Ilm. INTI*, vol. 3, no. 1, pp. 100–104, 2014.
[6] M. Zarlis and Handrizal, *Algoritma & Pemrograman : Teori dan Praktik dalam Pascal*. Medan, 2008.
[7] T. Gutman, E. Aan, and C. Funny Farady, "IMPLEMENTASI ALGORITMA ZHU- TAKAOKA PADA APLIKASI KAMUS ISTILAH MUSIK BERBASIS ANDROID Gutman," *J. Rekursif*, vol. 5, no. 2, pp. 147–153, 2017.
[8] Nasruddin Safaat H, *Pemograman Aplikasi Mobile Smartphone Dan Tablet PC Berbasis Android*. Bandung: Informatika Bandung, 2015.
[9] R. A. . M.Shalahuddin, *Rekayasa Perangkat Lunak*. Yogyakarta: Modulo, 2014.