

## Implementasi Metode *Run Length Encoding* (RLE) untuk Kompresi Citra

Ahmad Jalaluddin<sup>1</sup>, Yuliana Melita<sup>2</sup>

<sup>1)</sup> Universitas Islam Lamongan

<sup>2)</sup> Sekolah Tinggi Teknik Surabaya

[Odden.85@gmail.com](mailto:Odden.85@gmail.com) , [ymp@stts.edu](mailto:ymp@stts.edu)

### ABSTRAKSI

*Algoritma 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 kemuncuilannya sedikit akan dikodekan dengan jumlah bit yang lebih panjang. Run Length Encoding Cocok untuk pemampatan citra yang memiliki kelompok pixel berderajat keabuan yang sama.*

**Kata kunci :** Citra, Kompresi, Run Length Encoding.

### I. Pendahuluan

#### 1. Latar Belakang

Data atau informasi tidak hanya disajikan dalam bentuk teks, tetapi juga dapat berupa gambar, audio (bunyi, suara, musik) dan video. Keempat macam data atau informasi ini sering disebut dengan multimedia. Era informasi teknologi saat ini tidak dapat dipisahkan dari multimedia.

Pada umumnya representasi citra digital membutuhkan memori yang besar. Semakin besar ukuran citra tentu semakin besar pula memori yang dibutuhkannya. Pada sisi lain, kebanyakan citra mengandung duplikasi data. Duplikasi data pada citra dapat berarti dua hal. Pertama, besar kemungkinan suatu pixel dengan pixel tetangganya memiliki intensitas yang sama, sehingga penyimpanan setiap pixel memboroskan tempat. Kedua, citra banyak mengandung bagian (region) yang sama, sehingga bagian yang sama ini tidak perlu dikodekan berulang kali karena redundan.

Saat ini, kebanyakan aplikasi menginginkan representasi citra dengan kebutuhan memori yang sesedikit mungkin. Kompresi citra (image compression) bertujuan meminimalkan kebutuhan memori untuk merepresentasikan citra digital. Prinsip umum yang digunakan dalam kompresi citra adalah mengurangi duplikasi data di dalam citra sehingga memori yang dibutuhkan untuk merepresentasikan citra menjadi lebih sedikit dari pada representasi citra semula.

Kompresi citra memberikan manfaat yang sangat besar dalam industri multimedia saat ini. Salah satunya adalah pada proses pengiriman data (data transmission) pada

saluran komunikasi data. Citra yang telah dikompresi membutuhkan waktu pengiriman yang lebih singkat dibandingkan dengan citra yang tidak dikompresi. Contohnya aplikasi pengiriman gambar lewat fax, videoconferencing, pengiriman data medis, pengiriman gambar dari satelit luar angkasa, pengiriman gambar via telepon genggam, download gambar dari internet, dan sebagainya. Selain pada proses pengiriman data, kompresi citra juga bermanfaat pada penyimpanan data (*data storing*) di dalam media sekunder (*storage*). Citra yang telah dikompresi membutuhkan ruang memori di dalam media storage yang lebih sedikit dibandingkan dengan citra yang tidak dikompresi.

Ada dua tipe utama kompresi citra, yaitu kompresi tipe lossless dan kompresi tipe lossy. Kompresi tipe lossy adalah kompresi dimana terdapat data yang hilang selama proses kompresi.

Akibatnya kualitas citra yang dihasilkan jauh lebih rendah daripada kualitas citra asli. Sementara itu, kompresi tipe lossless tidak menghilangkan informasi setelah proses kompresi terjadi, akibatnya kualitas citra hasil kompresi tidak menurun. Salah satu contoh dari kompresi tipe lossless adalah metode *Run Length Encoding*.

#### 2. Rumusan Masalah

- Bagaimana cara kerja dari Metode Run Length Encoding sebagai teknik kompresi citra
- Bagaimana tingkat efisiensi memori file hasil kompresi dengan menggunakan Metode Run Length

Encoding.

- c. Bagaimana implementasi dari Metode Run Length Encoding sebagai teknik kompresi citra dengan menggunakan software matlab.

### 3. Batasan Masalah

Metode Run Length Encoding diimplementasikan menggunakan Matlab 2010.

### 4. Tujuan

- a. Untuk mengetahui dan memahami cara kerja teknik kompresi citra dengan metode Run Length Encoding.
- b. Untuk mengetahui tingkat efisiensi memori file hasil kompresi citra dengan Metode Run Length Encoding.
- c. Untuk mengimplementasikan Metode Run Length Encoding sebagai teknik kompresi citra dengan menggunakan Matlab.

### 5. Metode Penelitian

- a. Studi Pustaka (Library Research)  
Studi Pustaka dilakukan dengan cara mempelajari teori-teori literatur dan buku-buku yang berhubungan dengan objek kajian sebagai dasar dalam penelitian, dengan tujuan memperoleh dasar teoritis gambaran dari apa yang dilakukan.

## 2. TINJAUAN PUSTAKA

### 1. Representasi Citra Digital

#### a. Citra Biner (monokrom)

Pada citra biner setiap titik bernilai 0 atau 1, masing-masing merepresentasikan warna tertentu. Contohnya adalah warna hitam bernilai 0 dan warna putih bernilai 1, pada standar citra yang ditampilkan di layar komputer, nilai biner ini berhubungan dengan ada tidaknya cahaya yang ditembakkan oleh electron gun yang terdapat dalam monitor komputer. Angka 0 menyatakan tidak ada cahaya, dengan demikian warna yang direpresentasikan adalah hitam. Untuk angka 1 terdapat cahaya, sehingga warna yang direpresentasikan adalah putih.

#### b. Citra Skala Keabuan (gray scale)

Citra skala keabuan memberi kemungkinan warna yang lebih banyak daripada citra biner, karena ada nilai-nilai lain diantara nilai minimum (biasanya = 0) dan nilai maksimumnya. Banyaknya kemungkinan nilai

dan nilai maksimumnya bergantung pada jumlah bit yang digunakan. Contohnya dalam skala keabuan 4 bit, maka jumlah kemungkinan nilainya adalah  $2^4 = 16$ , dan nilai maksimumnya adalah  $2^4 - 1 = 15$ .

Format citra ini disebut skala keabuan karena pada umumnya warna yang dipakai adalah antara hitam sebagai warna minimal dan putih sebagai warna maksimal, sehingga warna antaranya adalah abu-abu.

#### c. Citra Warna (true colour)

Pada citra warna setiap titik mempunyai warna yang spesifik yang merupakan kombinasi dari tiga warna dasar, yaitu: merah, hijau, biru. Format citra ini sering disebut sebagai citra RGB (red- green-blue). Setiap warna dasar mempunyai intensitas sendiri dengan nilai maksimum 255 (8 bit).

Dengan demikian setiap titik pada citra warna membutuhkan 3 byte.

Jumlah kombinasi warna yang mungkin untuk format citra ini adalah  $2^{24}$  atau lebih dari 16 juta warna, dengan demikian dianggap mencakup semua warna yang ada, inilah sebabnya format ini dinamakan true colour.

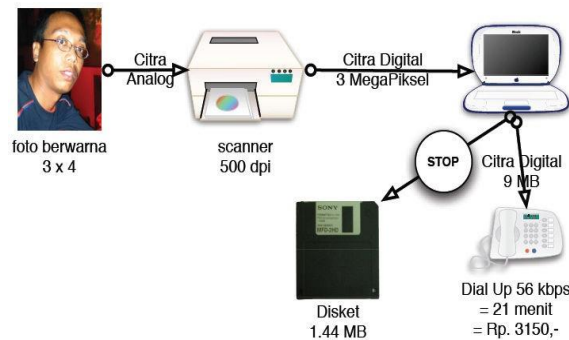
#### d. Citra Warna Berindeks

Jumlah memori yang dibutuhkan untuk format citra warna true colour adalah tiga kali jumlah titik yang ada dalam citra yang ditinjau. Di lain pihak pada kebanyakan kasus, jumlah warna yang ada dalam suatu citra terkadang sangat terbatas (jauh dibawah 10 juta kemungkinan warna yang ada) karena banyaknya warna dalam sebuah citra tidak mungkin melebihi banyaknya titik dalam citra itu sendiri. Untuk kasus tersebut disediakan format citra warna berindeks. Pada format ini, informasi setiap titik merupakan indeks dari suatu tabel yang berisi informasi warna yang tersedia yang disebut palet warna (beberapa buku menyebutnya sebagai colour map).

### 1. 2. Kompresi Citra

Kompresi citra bertujuan untuk meminimalkan jumlah bit yang diperlukan untuk merepresentasikan citra. Apabila sebuah foto berwarna berukuran 3 inci x 4 inci diubah ke bentuk digital dengan tingkat resolusi sebesar 500 dot per inch (dpi), maka diperlukan  $3 \times 4 \times 500 \times 500 = 3.000.000$  dot ( piksel). Setiap piksel terdiri dari 3 byte dimana masing-

masing byte merepresentasikan warna merah, hijau, dan biru. sehingga citra digital tersebut memerlukan volume penyimpanan sebesar  $3.000.000 \times 3 \text{ byte} + 1080 = 9.001.080 \text{ byte}$  setelah ditambahkan jumlah byte yang diperlukan untuk menyimpan format (header) citra.



**Gambar 1. Proses Konversi citra analog ke citra digital dan pengirimannya**

Parameter-parameter citra yang penting dalam proses kompresi diantaranya sebagai berikut:

#### 1. Resolusi

Resolusi citra menyatakan ukuran panjang kali lebar dari sebuah citra. Resolusi citra biasanya dinyatakan dalam satuan piksel. Semakin tinggi resolusi sebuah citra, semakin baik kualitas citra tersebut. Namun, tingginya resolusi menyebabkan semakin banyaknya jumlah bit yang diperlukan untuk menyimpan dan mentransmisikan data citra tersebut.

#### 2. Kedalaman bit

Kedalaman bit menyatakan jumlah bit yang diperlukan untuk merepresentasikan tiap piksel citra pada sebuah frame. Kedalaman bit biasanya dinyatakan dalam satuan bit/piksel. Semakin banyak jumlah bit yang digunakan untuk merepresentasikan sebuah citra, maka semakin baik kualitas citra tersebut.

#### 3. Konsep redundansi

Redundansi merupakan suatu keadaan dimana representasi suatu elemen data tidak bernilai signifikan dalam merepresentasikan keseluruhan data. Keadaan ini menyebabkan data keseluruhan dapat direpresentasikan secara lebih kompak dengan cara menghilangkan representasi dari sebuah elemen data yang redundan. Redundansi yang terdapat pada citra statik adalah redundansi

spasial.

Dalam proses kompresi (pemampatan) citra terdapat dua proses utama yaitu sebagai berikut:

1. Pemampatan citra (image compression)  
Pada proses, ini citra dalam representasi tidak mampat dikodekan dengan representasi yang meminimumkan kebutuhan memori.
2. Penirmampatan citra (image decompression)  
Pada proses ini, citra yang sudah dimampatkan harus dapat dikembalikan lagi (decoding) menjadi representasi yang tidak mampat. Proses ini diperlukan jika citra tersebut akan ditampilkan ke layar atau disimpan ke dalam arsip dengan format tidak mampat. Saat ini sudah banyak ditemukan metode-metode pemampatan citra. Kriteria yang digunakan dalam mengukur metode pemampatan citra adalah [6]:

1. Waktu pemampatan dan penirmampatan (*decompression*)
2. Kebutuhan memori
3. Kualitas pemampatan (*fidelity*)
4. Format keluaran

Ada empat jenis pendekatan dalam pemampatan citra adalah sebagai berikut:

1. Pendekatan statistik
2. Pendekatan ruang
3. Pendekatan kuantisasi
4. Pendekatan fractal

Metode pemampatan citra dapat diklasifikasikan ke dalam dua kelompok besar yaitu:

#### 1. Metode *lossless*

Metode lossless selalu menghasilkan citra hasil penirmampatan yang tepat sama dengan citra semula, pixel per pixel. Tidak ada informasi yang hilang akibat pemampatan. Contoh metode lossless adalah metode Run Length Encoding. Rasio kompresi citra dihitung dengan rumus:

$$\text{Rasio kompresi} = 100\% - \left( \frac{\text{ukuran citra semula}}{\text{ukuran citra hasil pemanfaatan}} \right) \times 100\%$$

Metode lossless cocok untuk memampatkan citra yang mengandung informasi penting yang tidak boleh rusak akibat pemampatan. Misalnya memampatkan gambar hasil diagnosa medis.

#### 2. Metode *lossy*

Metode lossy menghasilkan citra hasil pemampatan yang hampir sama dengan citra semula. Ada informasi yang hilang akibat

pemampatan, tetapi dapat ditolerir oleh persepsi mata. Mata tidak dapat membedakan perubahan kecil pada gambar. Metode pemampatan lossy menghasilkan nisbah pemampatan yang tinggi daripada metode lossless.

### 3. Metode *Run Length Encoding*

Algoritma RLE ini merupakan teknik kompresi yang lossless, sehingga kita dengan mudah dapat melakukan dekompresi citra kembali. Bagaimana algoritma RLE bekerja? Kita coba implementasikan dengan citra grayscale ukuran 6x6 bit (8 derajat keabuan) sebagai berikut:

```
1 1 1 7 1 3
4 4 6 1 2 2
7 7 7 5 5 5
6 4 4 2 2 2
5 5 2 2 2 1
2 3 3 3 0 0
```

Kode RLE merupakan pasangan intensitas warna dan banyaknya intensitas warna yang muncul berurutan. Sehingga di dapat kode RLE nya adalah sebagai berikut:

```
1 3 7 1 1 1 3 1 4 2 6 1 1 1 2 2 7 3 5 3 6 1 4 2 2 3
5 2 2 3 1 1 2 1 3 3 0 2
```

Cara membaca kode diatas adalah : intensitas warna 1 muncul 3 kali, intensitas warna 7 muncul 1 kali, intensitas warna 1 muncul 1 kali, intensitas warna 3 muncul 1 kali,..... dan seterusnya. Cocokkan dengan citra aslinya.

- RLE merupakan metode kompresi yang banyak didukung oleh format file gambar seperti: TIFF, BMP, PCX.
- RLE bekerja dengan mengurangi ukuran fisik dengan adanya pengulangan string dari deretan karakter / byte data.
- String perulangan ini dinamakan RUN, dan biasanya di kodekan dalam 2 byte. Byte pertama merupakan jumlah perulangan dan byte kedua adalah karakter yang diulang.

Cocok untuk pemampatan citra yang memiliki kelompok pixel berderajat keabuan yang sama

Contoh citra 10x10 dengan 8 derajat keabuan

0	0	0	0	0	2	2	2	2	2
0	0	0	1	1	1	1	2	2	2
1	1	1	1	1	1	1	1	1	1
4	4	4	4	3	3	3	3	2	2
3	3	3	5	5	7	7	7	7	6
2	2	6	0	0	0	0	1	1	0
3	3	4	4	3	2	2	2	1	1
0	0	0	0	0	0	0	0	1	1
1	1	1	1	0	0	0	2	2	2
3	3	3	2	2	2	1	1	1	1

```
(0, 5), (2, 5)
(0, 3), (1, 4), (2, 3)
(1, 10)
(4, 4), (3, 4), (2, 2)
(3, 3), (5, 2), (7, 4), (6, 1)
(2, 2), (6, 1), (0, 4), (1, 2), (0, 1)
(3, 2), (4, 2), (3, 1), (2, 2), (1, 2)
(0, 8), (1, 2)
(1, 4), (0, 3), (2, 3)
(3, 3), (2, 3), (1, 4)
```

Pasangan derajat keabuan ( $p$ ) dan jumlah pixel ( $q$ )

Ukuran citra sebelum dimampatkan (1 derajat keabuan = 3 bit) adalah 100 x 3 bit = 300 bit

Ukuran citra setelah pemampatan (run length =4) adalah (31 x 3) + (31 x 4) bit = 217 bit

Nisbah pemampatan =  $(100\% - \frac{217}{300} \times 100\%) = 27.67\%$ , yang artinya 27.67%

### 3. IMPLEMENTASI

1. Potongan Kode program untuk uji coba kompresing pada sebuah citra.

```
function Output=rle(Input)
L=length(Input);
j=1;
k=1;
i=1;
while i<2*L
    comp=1;
    for j=j:L
        if j==L
            break
        end;
        if Input(j)==Input
            (j+1)
            comp=comp+1;
        else
            break
        end;
    end;
    Output(k+1)=comp;
    Output(k)=Input(j);
```

```

    if j==L && Input
(j-1)==Input(j)
    break
end;
i=i+1;
k=k+2;
j=j+1;
if j==L
    if mod(L,2)==0
        Output(k+1)=1;
        Output(k)=Input
(j);
    else
        Output(k
+1)=1;
        Output(k)=Input
(j);
    end;
    break
end;
end;
end;

```

## 2 .Hasil Kompresi Citra

a.Citra Asli



ukuran citra asli sebesar 91

b.Hasil Compresing Citra



ukuran citra setelah dikompres  
menjadi 80 Kb

g adalah citra grayscale yang kita punya. Hasilnya akan ditampung dalam array yang diberi nama rle. Nah karena dapam array rle kita juga menyimpan berapa kali intensitas warna tersebut muncul, maka untuk mendekompresi kembali tentu merupakan perkara yang mudah.

## 4. PENUTUP

### 1. Kesimpulan

Kompresi terhadap citra RGB dilakukan dengan cara menurunkan intensitas tingkat R, G dan B secara seragam. Penurunan tersebut mengakibatkan bit yang merepresentasikan citrapun menjadi berkurang. Karena bit yang merepresentasikan citra berkurang, maka besar file citra pun akan semakin kecil.

1. Teknik kompresi citra dengan metode

Run Length Encoding pada citra gray scale 2. Kode biner yang dihasilkan dari Metode Run Length Encoding adalah setiap nilai piksel yang mempunyai frekuensi terbesar mempunyai jumlah bit yang pendek sedangkan nilai piksel yang mempunyai frekuensi kecil mempunyai bit yang lebih panjang.

2. Citra yang mempunyai sebaran nilai piksel tidak merata memiliki rasio kompresi yang relatif besar sedangkan citra dengan nilai piksel yang merata memiliki rasio kompresi yang lebih kecil.
3. Tingkat efisiensi memori file hasil kompresi dengan menggunakan Metode Run Length Encoding diukur dari besarnya rasio kompresi yang dihasilkan. Citra yang mempunyai sebaran nilai piksel tidak merata memiliki tingkat efisiensi lebih besar dibandingkan dengan citra dengan nilai piksel yang merata.

## 2. Saran

Untuk dapat lebih melihat dan membuktikan keefektifan, kelebihan dan kelemahan dari algoritma Run Length Encoding, perlu diadakannya sebuah penelitian yang bertujuan membandingkan seluruh algoritma kompresi dalam mengompres berbagai citra.

## DAFTAR PUSTAKA

Run Length Encoding Coding [http://www.en.wikipedia.org/wiki/Run\\_Length\\_Encoding\\_coding](http://www.en.wikipedia.org/wiki/Run_Length_Encoding_coding), Tanggal akses : Tanggal akses: Sabtu, 15 Maret 2008 pukul 06:28

Munir, Rinaldi. 2004. *Pengolahan Citra Digital dengan pendekatan Algoritmik*. Informatika, Bandung.

Munir, Rinaldi. 2005. *Algoritma dan Pemrograman dalam bahasa Pascal dan C* Edisi ke-3. Informatika, Bandung

Halaman ini sengaja dikosongkan