

# Rancang Bangun Aplikasi Deteksi Kemiripan Dua Gambar Menggunakan Algoritma Ratcliff/Obershelp

Lina Wati Aritonang

Program Studi Teknik Informatika, STMIK Budi Darma, Medan, Indonesia

Email: lina.aritonang@gmail.com

**Abstrak**—Pencarian gambar dengan tingkat kemiripan dua buah gambar pada sekumpulan file dengan algoritma Ratcliff/Obershelp sangat diharapkan untuk membantu pekerjaan manusia. Algoritma ini merupakan salah satu algoritma yang bisa digunakan untuk mencari kemiripan dua buah gambar dengan mengambil nilai rgb dari masing-masing rgb dari pixel tiap gambar. Proses pencarian untuk mendapatkan gambar dengan kemiripan tertinggi. Format gambar yang bisa diinputkan kedalam sistem adalah citra bitmap. Pemetaan bitmap gambar grafis komputer yang terdiri atas titik-titik yang membentuk baris dan kolom, citra yang terdiri atas titik dan pixel. Pada format bitmap, citra disimpan sebagai sebagai suatu matriks dimana masing-masing elemennya digunakan untuk menyimpan informasi warna untuk menyimpan informasi warna untuk setiap pixel. Penelitian ini menggunakan algoritma Ratcliff/Obershelp merupakan algoritma pencarian citra yang dilakukan dengan pencarian pencocokan dari nilai rgb yang sama pada citra. Proses pencarian kemiripan gambar dapat membantu mengatasi masalah pada tingkat kemiripan gambar dan pencarian gambar lebih cepat dan akurat. Algoritma Ratcliff/Obershelp membantu pencarian kemiripan gambar dapat diterapkan sehingga proses pencarian dapat lebih mudah. Algoritma Ratcliff/Obershelp dilakukan secara detail dengan menampilkan nilai query dari tingkat kemiripan gambar dan bagian yang dieksekusinya serta menampilkan gambar dengan format bmp, jpg dan png terhadap pencarian kemiripan gambar.

**Kata Kunci:** File Teks, Kompresi, Algoritma Elias Delta Code, Algoritma Levenstein

**Abstract**— Image search with the level of similarity of two images in a set of files with the Ratcliff / Obershelp algorithm is expected to help human work. This algorithm is an algorithm that can be used to find the similarity of two images by taking the RGB value of each RGB of the pixels per image. The process of searching for images with the highest likeness. Image format that can be input into the system is a bitmap image. Bitmap mapping of computer graphic images consisting of dots that form rows and columns, images consisting of dots and pixels. In the bitmap format, the image is stored as a matrix where each element is used to store color information to store color information for each pixel. This study uses the Ratcliff / Obershelp algorithm which is an image search algorithm which is performed by searching for matching of the same RGB value in the image. The image similarity search process can help resolve problems at the image similarity level and image search faster and more accurately. Ratcliff / Obershelp's algorithm helps image similarity search can be applied so that the search process can be easier. The Ratcliff / Obershelp algorithm is performed in detail by displaying query values from the level the similarity of the image and the part that it executes and displays the image in bmp, jpg and png format to the image similarity search.

**Keywords:** T Search, Image, Image Similarity, Ratcliff/Obershelp

## 1. PENDAHULUAN

Pencarian kemiripan antara dua buah gambar bukanlah hal yang bisa dengan mudah dilakukan oleh sebuah komputer. Padahal, keberadaan komputer diharapkan bisa mempermudah pekerjaan manusia dengan lebih cepat dan akurat termasuk didalamnya untuk mencari kemiripan antara dua buah gambar. Oleh karena itu, diperlukan sebuah solusi yang tepat untuk memecahkan masalah. Seiring dengan berkembangnya ilmu komputer, dihasilkan beberapa metode yang bisa digunakan sebagai solusi untuk memecahkan masalah pencarian kemiripan antara dua buah gambar.

Masalah yang dihadapi pada deteksi suatu proses untuk mencari atau melakukan pencarian terhadap sesuatu dengan menggunakan cara dan teknik tertentu. Deteksi kemiripan dua buah gambar dapat digunakan untuk berbagai masalah, dimana sistem mengidentifikasi masalah-masalah yang berhubungan dengan pendeteksian kemiripan dua buah gambar. Akan tetapi dengan alat tersebut sudah pasti akan membuat berkas gambar yang di miliki oleh seorang user akan sangat banyak dan kemungkinan besar berkas tersebut adalah *duplikat atau sama (kemiripan gambar)*.

Algoritma *Ratcliff/Obershelp* menggunakan proses yang sama untuk memutuskan seberapa mirip dua pola satu dimensi. Karena *string* piksel merupakan satu dimensi, algoritma ini mengembalikan nilai yang dapat di gunakan sebagai faktor kepercayaan atau persentase, menunjukkan kesamaan dua *string*. Konsep pencocokan dari algoritma ini yaitu, pertama menemukan sub string terpanjang yang memiliki kesamaan dari *string* dalam kemiripan dua buah gambar di sebut *anchor*. Kemudian bagian yang tersisa dari *string* sebelah kiri dan kanan dari *anchor* harus diperiksa sebagai *string-string* yang baru (dengan kata lain mengulangi step 1). Proses tersebut di ulangi sampai semua karakter dari string kemiripan pada dua buah gambar di analisa.[1]

*Yudhy Lady Joane, Alicia Sinsuw, Agustinus Jacobus*” Rancang Bangun Aplikasi Deteksi Kemiripan Dokumen Teks Menggunakan Algoritma Ratcliff/Obershelp” Junal Teknik Informatika Vol. 11, No.1, 2017 ISSN : 2301-8364. Kegiatan menjiplak tugas sering dilakukan mahasiswa yang merupakan tindakan plagiat, banyak tugas yang terkumpul dan waktu yang terbatas membuat dosen sulit untuk memeriksa tugas satu per satu. Oleh karena itu, diperlukan suatu aplikasi yang dapat mendeteksian kemiripan dokumen teks. Penelitian ini

menggunakan algoritma *Ratcliff/Obershelp* yang dapat menilai seberapa mirip pola dari dua string yang berbeda[2].

Dengan penerapan Algoritma *Ratcliff/Obershelp* pada sebuah aplikasi, maka kemungkinan besar masalah penduplikasian gambar-gambar yang ada dapat teratasi dengan baik. Yaitu dengan memindai semua berkas gambar yang ada di dalam tempat penyimpanan dan membandingkan semua berkas gambar tersebut hingga mendapatkan hasil. Nantinya nilai perbedaan terkecil dari hasil pengembalian citra akan menjadi tanda bahwa gambar-gambar tersebut mempunyai berkas yang sama atau mirip pada berkas yang lainnya, dan pada akhirnya user atau pemilik berkas dapat mengambil tindakan untuk menghapus atau tindakan lain.

## 2. METODOLOGI PENELITIAN

### 2.1 Information Retrieval

*Information Retrieval* merupakan studi tentang sistem pengindeksan, pencarian, dan mengingat data, khususnya teks atau bentuk tidak terstruktur lainnya yang dapat memenuhi informasi dari dalam koleksi besar yang biasanya disimpan dalam komputer.[4]. *Information Retrieval* merupakan suatu sistem yang menemukan informasi yang sesuai dengan kebutuhan *user* dari kumpulan informasi secara otomatis. *Information Retrieval* sering digunakan untuk aplikasi pencarian contohnya seperti mesin pencarian internet.

Sistem temu balik informasi adalah suatu sistem yang mampu melakukan penyimpanan, pencarian, dan pemeliharaan informasi. Informasi dalam konteks ini dapat terdiri dari gambar (termasuk data numerik dan tanggal), gambar, audio, video, dan objek multimedia lainnya.[5]

### 2.2 Analisis Citra

Analisis citra bertujuan untuk mengidentifikasi parameter-parameter yang diasosiasikan dengan ciri dari objek didalam citra, untuk selanjutnya parameter tersebut digunakan dalam menginterpretasi citra. Analisis citra pada dasarnya terdiri dari tiga tahapan yaitu ekstraksi ciri, segmentasi dan klasifikasi. Faktor kunci dalam mengekstraksi ciri adalah kemampuan mendeteksi keberadaan tepi dari objek didalam citra. Setelah tepi objek diketahui, langkah selanjutnya dalam analisis citra adalah segmentasi, yaitu me reduksi citra menjadi objek atau region. Langkah terakhir dari analisis citra adalah klasifikasi, yaitu memetakan segmen-segmen yang berbeda kedalam kelas objek yang berbeda pula. [6].

### 2.3 Algoritma Ratcliff/Obershelp

Algoritma *Ratcliff/Obershelp* menggunakan proses yang sama untuk memutuskan seberapa mirip dua pola satu dimensi. Karena string teks merupakan satu dimensi, algoritma ini mengembalikan nilai yang dapat di gunakan sebagai faktor kepercayaan atau persentase, menunjukkan kesamaan dua string[2]. Konsep pencocokan dari algoritma ini yaitu, pertama menemukan sub string terpanjang yang memiliki kesamaan dari string S1 dan S2 yang di sebut *anchor*. Nilai dari Km bertambah berdasarkan panjang dari *anchor*. Kemudian bagian yang tersisa dari string sebelah kiri dan kanan dari *anchor* harus diperiksa sebagai string-string yang baru (dengan kata lain mengulangi step 1). Proses tersebut di ulangi sampai semua karakter dari string S1 dan S2 di analisa. Algoritma *Ratcliff/Obershelp* dinyatakan dengan rumus :

$$D_{ro} = \frac{2 * Km}{|S1| + |S2|}$$

Keterangan :

Km = Jumlah karakter yang sama

|S1| = Panjang dari string 1

|S2| = Panjang dari string 2

## 3. HASIL DAN PEMBAHASAN

### 3.1 Analisa Masalah











Analisis Aplikasi pencarian kemiripan gambar bertujuan untuk mempermudah menemukan file gambar dan membandingkan gambar-gambar yang sama atau mirip. Permasalahan yang dihadapi adalah bagaimana sistem dapat menemukan kemiripan pada semua file gambar yang berformat Jpeg. Setelah mendapatkan file gambar yang dimaksud, pengguna dapat melihat tingkat kemiripan pada gambar yang sama dan memilih mode pencarian dengan menerapkan *Ratcliff/Obershelp*, merupakan sebuah algoritma untuk mencari sub string terpanjang yang memiliki kesamaan dari *string*. Akan tetapi algoritma ini akan digunakan untuk mencari kemiripan sebuah gambar dengan gambar lain yang ada di sekumpulan gambar dan kemungkinan besar berkas tersebut adalah *duplikat atau sama (kemiripan gambar)*. Dan masalah ini sangat sulit untuk diatasi jika dikerjakan secara manual contoh, untuk menghapus gambar-gambar yang sama, harus memilah dan mengingat gambar tersebut kemudian menghapusnya. Oleh sebab itu bagaimana merancang aplikasi yang dapat menemukan kemiripan gambar yang sama berdasarkan

algoritma *Ratcliff/Obershelp*. Aplikasi yang dibuat nantinya dapat membantu dan mempermudah pengguna untuk mencari gambar yang mirip secara akurat.

### 3.2 Penerapan Algoritma *Ratcliff/Obershelp*

Permasalahan yang akan diangkat dari penerapan ini adalah bagaimana mengklasifikasikan pada Algoritma *Ratcliff/Obershelp* pada citra. Format *file* citra didalam penelitian ini adalah citra berwarna. Berikut ini merupakan contoh penggunaan gambar dapat dilihat pada tingkat kemiripan dengan format bmp.

Tabel 1. Kemiripan Gambar

Gambar 1	Gambar 2	Resolusi bitmap ( <i>bmp</i> )	
		296x170	296x178
		296x170	296x178
		8x8	8x8
		269x187	284x177
		259x194	300x168

Dari gambar-gambar diatas memiliki segmentasi warna. Pada gambar diatas penulis mengambil sample dari gambar ke satu dan gambar ke dua dengan mobil berwarna merah dengan resolusi 8x8 piksel berformat bmp. Dari gambar ke satu mobil berwarna merah, memperlihatkan matriks pada nilai-nilai piksel setiap baris dan kolom adalah sebagai berikut :



Gambar 1. Citra 8x8 piksel

Berikut ini merupakan nilai ASCII yang di ambil dari setiap baris pertama nilai RGB  
Perintah untuk mengambil nilai warna R :

Ascii (,:,1) =

R	232	221	209	189	162	141	126	123
	239	219	200	176	154	139	131	129
	246	216	186	160	146	141	139	138
	232	198	161	136	129	134	137	137
	205	170	135	113	112	124	132	133
	173	141	111	95	99	115	125	129
	148	120	94	80	87	107	117	121
	143	116	91	77	86	105	117	120

Perintah untuk mengambil nilai warna G :

Ascii (,:,2) =

	247	234	219	193	163	140	124	121
	250	229	205	177	152	134	123	121
	250	220	186	155	134	127	123	122
	236	200	159	128	114	116	116	116
	210	173	134	107	100	106	110	112
	182	149	113	90	89	99	106	108

G	163	132	99	80	79	93	100	102
	161	131	99	79	80	93	101	103

Perintah untuk mengambil nilai warna B :

Ascii (,:,3) =

B	216	206	195	176	155	136	127	126
	220	202	182	159	140	128	121	119
	223	193	162	135	118	114	110	109
	211	176	136	105	93	96	97	95
	190	152	113	85	76	82	86	85
	163	128	92	68	64	73	76	77
	142	110	77	56	56	67	72	72
	139	108	76	55	56	67	75	75

Pada contoh kasus ini dilakukan pada algoritma *Ratcliff/Obershelp* mencari kesamaan kedua string dari nilai RGB terhadap citra dengan ukuran 8x8 piksel :

R		1	2	3	4	5	6	7	8
	S1	232	221	209	189	162	141	126	123
	S2	239	219	200	176	154	139	131	129
	<b>S1</b>	246	216	186	160	146	141	139	138
	<b>S2</b>	232	198	161	136	129	134	137	137
	S1	205	170	135	113	112	124	132	133
	S2	173	141	111	95	99	115	125	129
	S1	148	120	94	80	87	107	117	121

1. Panjang dari string S1 : |S1|=8  
Panjang dari string S2 : |S2|=8
2. Substring yang terpanjang yang dimiliki oleh kedua string pada nilai Red adalah 141, 137. Maka nilai merupakan sebuah *anchor*, dan  $Km = |141, 137| = 2$

R		1	2	3	4	5	6	7	8
	S1	246	216	186	160	146	141	139	138
	S2	232	198	161	136	129	134	137	137

3. Di sebelah kiri dari *anchor* tersisa kumpulan nilai 232, 198, 161, 136, 129, 134 dan 246, 216, 186, 160, 146. Sub string yang terpanjang dan memiliki kesamaan dari kumpulan nilai red tersebut adalah 141, 137. Maka,  $Km = 2 + |141, 137| = 280$

R		1	2	3	4	5	6	7	8
	S1	246	216	186	160	146	141	139	138
	S2	232	198	161	136	129	134	137	137

4. Nilai 246, 216, 186, 160, 146 dan 232, 198, 161, 136, 129, 134 merupakan substring yang berada di awal dari kedua string S1 dan S2, disebelah kiri substring tersebut sudah tidak terdapat nilai lagi. Pada sebelah kanan, memiliki nilai 139 dan 138. Maka Km tetap sama dan kita lanjutkan pada nilai sebelah kanan dari *anchor*.

Penilaian *Ratcliff/Obershelp* untuk string pada nilai red adalah:

$$D_{ro} = \frac{2 \times 280}{|8| + |8|} = \frac{560}{16} = 35,00$$

Jadi, dari kedua string nilai Red memiliki nilai kesamaan : 35,00

G		1	2	3	4	5	6	7	8
	S1	247	234	219	193	163	140	124	121
	S2	250	229	205	177	152	134	123	121
	<b>S1</b>	250	220	186	155	134	127	123	122
	<b>S2</b>	236	200	159	128	114	116	116	116
	S1	210	173	134	107	100	106	110	112
	S2	182	149	113	90	89	99	106	108
	S1	163	132	99	80	79	93	100	102

1. Panjang dari string S1 : |S1|=8  
Panjang dari string S2 : |S2|=8
2. Substring yang terpanjang yang dimiliki oleh

kedua string pada nilai Green adalah 250, 134, 123, 116. Maka nilai merupakan sebuah *anchor*, dan  $Km = |250, 134, 123, 116| = 4$

G	S1	250	220	186	155	134	127	123	122
	S2	236	200	159	128	114	116	116	116

- Di sebelah kiri dari *anchor* tersisa kumpulan nilai 236, 200, 159, 128, 114 dan 220, 186, 155. Sub string yang terpanjang dan memiliki kesamaan dari kumpulan nilai green tersebut adalah 250, 134, 123, 116. Maka,  $Km = 4 + |250, 134, 123, 116| = 623$

G	S1	250	220	186	155	134	127	123	122
	S2	236	200	159	128	114	116	116	116

- Nilai 220, 186, 155 dan 236, 200, 159, 128, 114 merupakan substring yang berada di awal dari kedua string S1 dan S2, disebelah kiri substring tersebut sudah tidak terdapat nilai lagi. Pada sebelah kanan, memiliki nilai 127 dan 122 di dalam string S1 dan pada string S2 sudah tidak memiliki nilai lagi. Maka Km tetap sama dan kita lanjutkan pada nilai sebelah kiri dari *anchor*.

Penilaian *Ratcliff/Obershelp* untuk string pada nilai green adalah:

$$D_{ro} = \frac{2 \times 623}{|18| + |8|} = \frac{1246}{16} = 77,87$$

Jadi, dari kedua string nilai Green memiliki nilai kesamaan : 77,87

B

	1	2	3	4	5	6	7	8
S1	216	206	195	176	155	136	127	126
S2	220	202	182	159	140	128	121	119
S1	223	193	162	135	118	114	110	109
S2	211	176	136	105	93	96	97	95
S1	190	152	113	85	76	82	86	85
S2	163	128	92	68	64	73	76	77
S1	142	110	77	56	56	67	72	72

- Panjang dari string S1 :  $|S1| = 8$   
Panjang dari string S2 :  $|S2| = 8$
- Substring yang terpanjang yang dimiliki oleh kedua string pada nilai Blue adalah 85, 76, 77. Maka nilai merupakan sebuah *anchor*, dan  $Km = |85, 76, 77| = 3$

B	S1	190	152	113	85	76	82	86	85
	S2	163	128	92	68	64	73	76	77

- Di sebelah kiri dari *anchor* tersisa kumpulan nilai 163, 128, 92, 68, 68, 64, 73 dan 190, 152, 113. Sub string yang terpanjang dan memiliki kesamaan dari kumpulan nilai blue tersebut adalah 85, 76, 77 Maka,  $Km = 3 + |85, 76, 77| = 241$

B	S1	190	152	113	85	76	82	86	85
	S2	163	128	92	68	64	73	76	77

- Nilai 190, 152, 113 dan 163, 128, 92, 68, 68, 64, 73 merupakan substring yang berada di awal dari kedua string S1 dan S2, disebelah kiri substring tersebut sudah tidak terdapat nilai lagi. Pada sebelah kanan, memiliki nilai 82 dan 86 di dalam string S1 dan pada string S2 sudah tidak memiliki nilai lagi. Maka Km tetap sama dan kita lanjutkan pada nilai sebelah kanan dari *anchor*.

Penilaian *Ratcliff/Obershelp* untuk string pada nilai blue adalah:

$$D_{ro} = \frac{2 \times 241}{|18| + |8|} = \frac{482}{16} = 30,12$$

Kemudian melakukan deteksi kemiripan pada gambar ke dua dengan mobil berwarna merah dengan resolusi 8x8 piksel berformat bmp :



Gambar 2. Citra 8x8 piksel

Berikut Algoritma *Ratcliff/Obershelp* pada citra mencari kesamaan kedua string dari nilai RGB :

	1	2	3	4	5	6	7	8
S1	205	136	129	134	137	124	132	133
S2	213	141	110	195	199	137	125	129
S1	200	198	161	136	129	134	128	137

S2	210	173	134	107	100	106	110	112
S1	173	141	111	90	89	116	133	136
S2	148	120	94	80	87	107	117	121
S1	113	98	56	35	72	28	114	117

1. Panjang dari string S1 : |S1|=8  
Panjang dari string S2 : |S2|=8
2. Substring 141, 137. yang terpanjang yang dimiliki oleh kedua string pada nilai Red adalah 141, 137. Maka nilai merupakan sebuah *anchor*, dan  $Km = |141, 137| = 2$

G	S1	213	141	110	195	199	137	125	129
	S2	200	198	161	136	129	134	128	137

3. Di sebelah kanan dari *anchor* tersisa kumpulan nilai 200, 198, 161, 136, 129, 134, 128 dan 213, 110, 195, 199, 125, 129. Sub string yang terpanjang dan memiliki kesamaan dari kumpulan nilai red tersebut adalah Maka,  $Km = 2 + |141, 137| = 280$

G	S1	213	141	110	195	199	137	125	129
	S2	200	198	161	136	129	134	128	137

4. Nilai 213, 110, 195, 199, 125, 129 dan 200, 198, 161, 136, 129, 134, 128 merupakan substring yang berada di awal dari kedua string S1 dan S2, disebelah kanan substring tersebut sudah tidak terdapat nilai lagi. Pada sebelah kiri, memiliki nilai 213 di dalam string S1 dan pada string S2 sudah tidak memiliki nilai lagi. Maka  $Km$  tetap sama dan kita lanjutkan pada nilai sebelah kanan dari *anchor*.  
Penilaian *Ratcliff/Obershelp* untuk string pada nilai red adalah:

$$D_{ro} = \frac{2 \times 280}{||8| + |8|} = \frac{560}{16} = 35,00$$

Jadi, dari kedua string nilai Red memiliki nilai kesamaan : 35,00

G		1	2	3	4	5	6	7	8
	S1	210	273	250	107	100	106	110	132
	S2	205	277	135	113	112	124	123	133
	S1	232	198	161	136	129	134	134	122
	S2	250	110	123	116	107	115	125	116
	S1	161	131	113	79	80	93	101	109
	S2	182	149	97	90	89	99	106	105
	S1	163	132	99	80	79	93	100	102

1. Panjang dari string S1 : |S1|=8  
Panjang dari string S2 : |S2|=8
2. Substring yang terpanjang yang dimiliki oleh kedua string pada nilai Green adalah 134, 123, 116, 250. Maka nilai merupakan sebuah *anchor*, dan  $Km = |134, 123, 116, 250| = 4$

B	S1	232	198	161	136	129	134	134	122
	S2	250	110	123	116	107	115	125	116

3. Di sebelah kiri dari *anchor* tersisa kumpulan nilai 110, 107, 115, 125 dan 232, 198, 161, 136, 129, 122. Sub string yang terpanjang dan memiliki kesamaan dari kumpulan nilai green tersebut adalah 134, 123, 116, 250. Maka,  $Km = 4 + |134, 123, 116, 250| = 623$

B	S1	232	198	161	136	129	134	134	122
	S2	250	110	123	116	107	115	125	116

4. Nilai 232, 198, 161, 136, 129, 122 dan 110, 107, 115, 125 merupakan substring yang berada di awal dari kedua string S1 dan S2, disebelah kiri substring tersebut sudah tidak terdapat nilai lagi. Pada sebelah kanan, memiliki nilai 122 di dalam string S1 dan pada string S2 sudah tidak memiliki nilai lagi. Maka  $Km$  tetap sama dan kita lanjutkan pada nilai sebelah kiri dari *anchor*.

Penilaian *Ratcliff/Obershelp* untuk string pada nilai green adalah:

$$D_{ro} = \frac{2 \times 623}{||8| + |8|} = \frac{1246}{16} = 77,87$$

Jadi, dari kedua string nilai Green memiliki nilai kesamaan : 77,87

B

	1	2	3	4	5	6	7	8
S1	223	293	162	135	118	114	110	129
S2	220	202	172	157	140	128	123	119
S1	123	170	142	130	150	104	120	110
S2	211	176	136	105	96	98	95	94
S1	190	152	113	87	76	84	86	85
S2	163	128	92	68	64	73	77	78
S1	142	110	77	53	56	85	76	72

1. Panjang dari string S1 : |S1|=8  
Panjang dari string S2 : |S2|=8
2. Substring yang terpanjang yang dimiliki oleh kedua string pada nilai Blue adalah 85, 76, 77. Maka nilai merupakan sebuah *anchor*, dan  $Km = |85, 76, 77| = 3$

G	S1	190	152	113	87	76	84	86	85
	S2	163	128	92	68	64	73	77	78

3. Di sebelah kanan dari *anchor* tersisa kumpulan nilai 163, 128, 92, 68, 64, 73, 78 dan 190, 152, 113, 87, 84, 86. Sub string yang terpanjang dan memiliki kesamaan dari kumpulan nilai blue tersebut adalah 85, 76, 77. Maka,  $Km = 3 + |85, 76, 77| = 241$



G	S1	190	152	113	87	76	84	86	85
	S2	163	128	92	68	64	73	77	78

4. Nilai 190, 152, 113, 87, 84, 86 dan 163, 128, 92, 68, 64, 73, 78 merupakan substring yang berada di awal dari kedua string S1 dan S2, disebelah kanan substring tersebut sudah tidak terdapat nilai lagi. Pada sebelah kiri, memiliki nilai 130 di dalam string S1 dan pada string S2 sudah tidak memiliki nilai lagi. Maka Km tetap sama dan kita lanjutkan pada nilai sebelah kanan dari *anchor*. Penilaian *Ratcliff/Obershelp* untuk string pada nilai blue adalah:

$$D_{ro} = \frac{2 \times 241}{|18| + |8|} = \frac{482}{16} = 30,12$$

Adapun hasil citra dari gambar ke satu dan ke dua pada gambar mobil berwarna merah memiliki hasil kemiripan dari nilai RGB dari penilaian *Ratcliff/Obershelp*. Dapat dilihat pada tabel 2. sebagai berikut :

**Tabel 2.** Hasil citra tingkat kemiripan

Gambar 1	resolusi (bmp)	Nilai kemiripan
	8x8	R = 82,75
		G = 47,37
		B = 72,75
		Kata-rata kemiripan % 202,87
Gambar 2	resolusi (bmp)	Nilai kemiripan
	8x8	R = 47,37
		G = 32,25
		B = 39,00
		Rata-rata kemiripan % 114,00

#### 4. KESIMPULAN

Berdasarkan hasil yang di dapat dalam penelitian dan penyusunan skripsi ini serta disesuaikan dengan tujuan, maka diperoleh kesimpulan sebagai berikut :

1. Proses pencarian kemiripan gambar dapat membantu mengatasi masalah pada tingkat kemiripan gambar dan pencarian gambar lebih cepat dan akurat.
2. Algoritma *Ratcliff/Obershelp* membantu pencarian kemiripan gambar dapat diterapkan sehingga proses pencarian dapat lebih mudah.
3. Algoritma *Ratcliff/Obershelp* dilakukan secara detail dengan menampilkan nilai query dari tingkat kemiripan gambar dan bagian yang dieksekusinya serta menampilkan gambar dengan format bmp, jpg dan png terhadap pencarian kemiripan gambar.

#### REFERENCES

- [1] Ilya Ilyankou, 2014. "Comparison of Jaro-Winkler and Ratcliff/Obershelp algorithms in spell check." IB Extended Essay Computer Science.

- [2] Yudhy Lady Joane, Alicia Sinsuw, Agustinus Jacobus” Rancang Bangun Aplikasi Deteksi Kemiripan Dokumen Teks Menggunakan Algoritma Ratcliff/Obershelp” Junal Teknik Informatika Vol. 11, No.1, 2017 ISSN : 2301-8364
- [3] Abdul kadir & Andi susanto, Teori dan Aplikasi Pengolahan Citra, Penerbit Andi, Yogyakarta : 2013
- [4] Nana ramadjanti, Modul Pratikum Color Histogram. Online Content Based Image Retrieval. Jakarta : PENS –ITS.
- [5] JURNAL TEKNIK POMITS. Ardian Yusuf Wicaksono, Implementasi Transformasi Curvelet dan Ruang Warna HSV untuk Temu Kembali Citra Batik Berbasis Isi pada Situs Batik. Vol. 2, No.1, 2013.
- [6] Darma putra, Pengolahan Citra Digital, Penerbit Andi, Yogyakarta : 2010
- [7] Aniati murni, Pengantar pengolahan citra, Elex Media Komputindo. Jakarta : 2007.
- [8] Jogiyanto.HM. 2005. Analisis dan Desain Sistem Informasi, ANDI. Yogyakarta
- [9] Adi Nugroho, 2010, Rekayasa Perangkat Lunak menggunakan UML dan Java, penerbit ANDI : Yogyakarta
- [10] Priyanto, Rahmat, 2009, Langsung Bisa Visual Basic.Net 2008 ,Penerbit ANDI, Yogyakarta.