

PENGUKURAN KEMIRIPAN FITUR PADA SISTEM TEMU KEMBALI CITRA BERBASIS KONTEN MENGGUNAKAN EUCLIDIAN DISTANCE

¹Rini Arianty, ²Maukar, ³Octarina Budi Lestari
^{1,2}Fakultas Ilmu Komputer Universitas Gunadarma
Jl. Margonda Raya No. 100, Depok 16424, Jawa Barat
¹rinia@staff.gunadarma.ac.id, ²maukar@staff.gunadarma.ac.id,
³octa_bl@staff.gunadarma.ac.id

Abstrak

Sistem Temu Kembali Citra berbasis Konten atau sistem, sudah diterapkan pada beberapa mesin pencari seperti Google dan Bing, tetapi citra hasil pencarian yang diberikan masih ada yang tidak relevan dengan citra permintaan. Membangun Temu Kembali Citra berbasis Konten yang dapat memberikan hasil pencarian yang relevan tergantung pada penarikan informasi dari konten citra yang dimasukkan. Proses penarikan informasi terhadap konten suatu citra dapat dilakukan dengan menggunakan metode ekstraksi fitur berdasarkan konten warna, bentuk atau tekstur. Penelitian ini, mengukur jarak kesamaan atau kemiripan antara citra query dengan citra pada database menggunakan Euclidian Distance pada Sistem Temu Kembali Citra berbasis Konten berdasarkan warna dan tekstur. Ekstraksi fitur warna dilakukan menggunakan metode Momen Warna, dan fitur tekstur menggunakan Filter Gabor. Persentase presisi tingkat keberhasilan Sistem yang diuji pada setiap kategori menggunakan pengujian secara visual dengan memperhatikan citra groundtruth. Hasil terendah memiliki presisi sejumlah 50% pada kategori gunung dan presisi tertinggi sejumlah 100% pada kategori dinosaurus. Rata-rata persentase presisi tingkat keberhasilan Sistem Temu Kembali Citra berbasis Konten sejumlah 84% dari 10 data uji yang diambil dari database. Hasil yang diharapkan dari penelitian, aplikasi dapat mengidentifikasi citra berdasarkan ekstraksi fitur yang digunakan dan dapat menampilkan 10 citra yang mirip dengan citra query pada perangkat desktop.

Kata Kunci: CBIR, Ekstraksi, Fitur, Tekstur, Warna.

Abstract

The Content Based Image Retrieval (CBIR) system or system has been applied to several search engines such as Google and Bing, however the image of the search results provided is occasionally irrelevant to the requested image. Bbuilding Content Based Image Retrieval that can provide relevant results depends on the retrieval of information from the image content entered. The process of retrieving information on the content of an image can be done using feature extraction method based on colour content, shape or texture. This research will be measured distance among the query image and the database image using Euclidian Distance in Content-based Image Retrieval System which is based on colour and texture content. The colour feature extraction was done using the Colour and Texture features using the Gabor Filter. The percentage of the precision level of success of the tested system in each category uses visual testing by paying attention to the ground truth image. The lowest results have the precision of 50% in the mountain category and the highest precision of 100% in the dinosaur category. The average percentage precision of the Content-based Image Retrieval System success rate is 84% of the 10 data taken from the database. The expected result of this research is the application can identify images based on the feature extraction used and it can show 10 images which are similar to the query image on the desktop device.

Keywords: CBIR, Extraction, Featue, Texture, Color.

PENDAHULUAN

Sistem temu kembali citra berbasis konten (*Content Based Image Retrieval / CBIR*) sudah diterapkan pada beberapa mesin pencari [1] seperti Google dan Bing, tetapi citra hasil pencarian yang diberikan masih ada yang tidak relevan dengan citra permintaan[2]. Proses membangun sistem *Content Based Image Retrieval (CBIR)* yang dapat memberikan hasil pencarian yang relevan tidak mudah[3]. Hasil pencarian yang relevan tergantung pada penarikan informasi dari konten citra [4] yang dimasukkan. Proses penarikan informasi terhadap konten suatu citra dapat dilakukan dengan menggunakan metode-metode ekstraksi fitur, fitur yang di ekstrak dapat berdasarkan konten warna, bentuk atau tekstur. Sistem *Content Based Image Retrieval (CBIR)* dapat digunakan lebih dari satu fitur [5] selama fitur itu dapat melengkapi satu sama lain [6]. Fitur yang telah selesai diekstrak akan dimasukkan ke dalam himpunan (*array*) yang disebut *feature vector* atau vektor fitur. Fitur-fitur tersebut yang akan digunakan nantinya pada proses pengukuran kemiripan fitur [7], untuk mendapatkan hasil citra yang mirip atau relevan dengan citra permintaan atau *query*. Berdasarkan penelitian terdahulu, penelitian terkait dengan sistem temu kembali citra berbasis konten ini sudah dilakukan oleh beberapa peneliti. Penelitian [8] mengimplementasikan CBIR untuk identifikasi citra berdasarkan warna dominan dengan menggunakan metode Transformasi

wavelet dan Histogram warna yang digabungkan, aplikasi berhasil menampilkan citra yang serupa dengan citra *query* berdasarkan informasi yang ada pada citra tersebut. Penelitian [9] menerapkan kombinasi fitur warna, bentuk, dan Siamse *Neural Network (SNN)* untuk mengembangkan sistem CBIR yang sebelumnya mereka bangun, Hasil yang diperoleh dari penelitian ini adalah nilai presisi rata-rata untuk semua kategori 98,62% yang melebihi dari karya yang sudah ada sebelumnya. Penelitian [10] menggunakan *Color Descriptor (CN)* dan *Gray Level Co-occurrence Matrix (GLCM)* untuk mengidentifikasi citra berdasarkan warna dan tekstur, Hasil yang didapatkan dari penelitian ini adalah metode untuk pengambilan citra menggunakan kombinasi fitur tekstur dan warna.

Penelitian [11] ini mengimplementasikan skema CBIR berbasis warna dan tekstur fitur visual, di mana fitur-fitur visual ini diekstraksi dari citra ruang warna RGB ke *Hue, Saturation, Value (HSV)* yang ditingkatkan menggunakan filter Laplacian, Hasil yang dicapai dalam penelitian ini adalah didapatkannya penarikan citra yang lebih relevan daripada beberapa skema lain yang ada, dengan nilai presisi rata-rata 89,50%. Penelitian [12] menerapkan metode *Active Re-Ranking* di mana citra permintaan yang telah disajikan dibangun modularnya, Hasil dari penelitian ini adalah pengambilan citra yang lebih baik setelah mengimplementasikan metode Active Re-Ranking. Pada penelitian ini

akan diimplementasikan penggunaan *Euclidean Distance* sebagai pengukur jarak kesamaan atau kemiripan antara citra *query* dengan citra pada *database* pada kumpulan citra di dalam *dataset* James Z. Wang [13] sistem temu kembali citra berbasis konten. Penelitian ini menggunakan citra RGB sejumlah 1000 buah citra dengan 10 kategori (antara lain: African, Pantai, Bangunan, Bis, Dinosaurus, Gajah, Bunga, Kuda, Gunung dan Makanan). Setiap kategori memiliki 100 buah citra. Ekstraksi fitur konten dilakukan berdasarkan warna dan tekstur menggunakan momen warna dan filter Gabor. Hasil yang diharapkan dari penelitian ini aplikasi dapat mengidentifikasi citra berdasarkan ekstraksi fitur yang digunakan dan dapat menampilkan citra yang relevan dengan citra *query* pada perangkat *desktop*.

METODE PENELITIAN

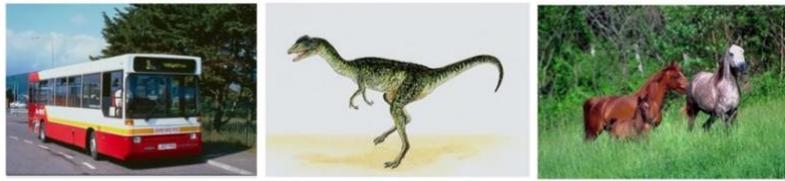
Pada penelitian ini dilakukan beberapa tahapan-tahapan dalam melakukan pengukuran kemiripan fitur pada sistem temukembali citra berbasis konten berdasarkan analisis warna dan tekstur.

1. Citra Masukan

Citra yang dimasukkan terbagi menjadi dua kelompok, kelompok pertama yaitu

kumpulan citra pada *dataset*, dan kelompok kedua adalah citra permintaan atau *query* yang telah dipilih dari *dataset* dengan deskripsi berikut :

1. Kumpulan citra atau gambar yang digunakan pada penelitian ini adalah citra berwarna dari *dataset* yang bernama “test images” milik James Z. Wang dan Jia Li [13]. Citra yang digunakan merupakan citra berwarna dengan model warna RGB dengan keseluruhan citra berjumlah 1000 buah citra yang terbagi menjadi 10 kategori, di mana pada setiap kategori memiliki 100 buah citra. Gambar 1 memperlihatkan beberapa contoh dari citra yang digunakan. Kumpulan citra ini nantinya akan diresize, lalu diekstrak fitur warna dan teksturnya, setelah itu *feature vector* yang dihasilkan pada setiap citra disimpan ke dalam file *.mat* yang nantinya akan diukur tingkat kemiripan fiturnya dengan citra *query*. Kategori-kategori citra yang terdapat pada dataset dapat dilihat pada Tabel 1.
2. Citra *query* merupakan citra yang akan dimasukkan sebagai permintaan [14] untuk dicari citra yang mirip. Citra *query* yang digunakan pada penelitian ini adalah citra pada *dataset* yang dipilih secara acak. Satu citra dipilih untuk setiap kategori.



Gambar 1. Contoh Citra yang Digunakan [13]

Tabel 1. Kategori Citra pada Dataset

No	Kategori	Gambar
1	African	0 - 99
2	Pantai	100 - 199
3	Bangunan	200 - 299
4	Bis	300 - 399
5	Dinosaurius	400 - 499
6	Gajah	500 - 599
7	Bunga	600 - 699
8	Kuda	700 - 799
9	Gunung	800 - 899
10	Makanan	900 - 999



(a). Citra Asli



(b) Citra *Resizing*

Gambar 2. Proses *Resizing*

2. Perubahan Ukuran Citra *Query* (*Resizing*)

Tahapan proses ini digunakan untuk mengubah ukuran citra *query* dan kumpulan citra pada *dataset* sebelum dilakukannya ekstraksi fitur. Citra diubah ukurannya menjadi ukuran standar 384 x 256 piksel, ukuran tersebut adalah ukuran citra yang

umum lebih banyak ada pada *dataset*. Berikut adalah *pseudocode* untuk proses *resizing*.

```
image = imresize(image, [384 256]);
```

Citra yang telah dimasukkan pada variabel “image” diubah ukurannya menjadi 384 x 256 dengan perintah “imresize”. Gambar 2 memperlihatkan ilustrasi perubahan ukuran citra asli yang sebelumnya berukuran 256 x

384 piksel, menjadi citra berukuran 384 x 256 piksel.

3. Ekstraksi Fitur

Ekstraksi fitur adalah proses mengambil ciri dari satu citra berdasarkan metode ekstraksi fitur yang digunakan [14][15]. Fitur yang diekstrak pada penelitian ini adalah warna dan tekstur. Metode yang digunakan untuk mengekstrak fitur warna adalah momen warna sedangkan metode yang digunakan untuk mengekstrak fitur tekstur adalah filter gabor. Pada tahap ini metode ekstraksi fitur dilakukan untuk mendapatkan fitur warna dan tekstur dalam bentuk array atau himpunan yang dinamakan *feature vector* [16]. *Feature vector* yang terbentuk dari momen warna dan filter gabor disatukan dan disimpan pada satu himpunan, untuk *feature vector* milik kumpulan citra pada *dataset* maka akan disimpan ke dalam file *.mat*, sedangkan *feature vector* milik citra *query* akan digunakan pada tahap pengukuran kemiripan fitur. Hasil akhir pada tahap ini adalah didapatkannya fitur warna dan tekstur yang sudah tersimpan pada file berekstensi *.mat* untuk fitur kumpulan citra pada *dataset*, dan siap diukur kemiripannya pada tahap selanjutnya.

A. Fitur Warna

Tahapan proses ini mengawali proses ekstraksi fitur, di mana fitur yang pertama kali diekstrak adalah fitur warna. Metode yang digunakan untuk ekstraksi citra berdasarkan

fitur warna pada tahap ini adalah momen warna. Ekstraktor warna ini menggunakan tiga momen utama dari distribusi warna citra yaitu *Mean* (Rata-Rata), *Standard deviation* (Deviasi standar), dan *Skewness* (Kecenderungan) [17]. Setelah proses perhitungan untuk ketiga momen utama selesai, maka akan menghasilkan tiga nilai untuk masing-masing komponen warna [18]. Hasil ketiga nilai ini merupakan fitur atau ciri dari warna citra yang telah dimasukkan. Fitur atau ciri ini disimpan pada suatu himpunan yang akan menjadi *feature vector* dari fitur warna. Berikut langkah dalam melakukan ekstraksi fitur warna menggunakan Momen Warna.

1. Ekstraksi saluran RGB pada citra menggunakan *pseudocode* Matlab berikut :

```
R = double(image(:, :, 1));  
G = double(image(:, :, 2));  
B = double(image(:, :, 3));
```

Pada sintaks tersebut dibuat tiga variabel untuk menampung nilai warna merah(*red*), hijau(*green*), dan biru(*blue*). Variabel “R” untuk warna merah, Variabel “G” untuk warna hijau, Variabel “B” untuk warna biru. Fungsi “*double*” digunakan untuk mengambil nilai matriks pada variabel “*image*”, di mana angka “1”, “2”, dan “3” menunjukkan matriks keberapa yang akan diambil datanya. Hasil dari proses ini akan menghasilkan tiga variabel yang berisi nilai dari masing-masing warna, dan data siap diproses untuk tahap selanjutnya.

- Menghitung nilai rata-rata pada setiap nilai warna merah(*Red*), hijau(*Green*), dan biru(*Blue*) menggunakan *pseudocode* Matlab berikut :

```
meanR = mean( R(:) );
```

Pada sintaks diatas variabel “R” yang berisi nilai warna merah dihitung nilai rata-ratanya menggunakan fungsi “mean”. Nilai dari proses perhitungan rata-rata tersebut disimpan pada variabel bernama “meanR”.

- Menghitung nilai deviasi standar pada setiap nilai warna merah(*Red*), hijau(*Green*), dan biru(*Blue*). Penghitungan nilai deviasi standar menggunakan *pseudocode* Matlab berikut:

```
stdR = std( R(:) );
```

Pada sintaks diatas variabel “R” yang berisi nilai warna merah dihitung nilai deviasi standarnya menggunakan fungsi “std”. Nilai dari proses perhitungan rata-rata tersebut disimpan pada variabel bernama “stdR”.

- Menghitung nilai kecenderungan atau *skewness* pada setiap nilai warna merah(*red*), hijau(*green*), dan biru(*blue*) menggunakan *pseudocode* Matlab berikut :

```
skR = skewness( R(:) );
```

Pada sintaks diatas variabel “R” yang berisi nilai warna merah dihitung nilai kecenderungannya menggunakan fungsi “std”. Nilai dari proses perhitungan

kecenderungan tersebut disimpan pada variabel bernama “skR”.

- Membuat himpunan *feature vector*. *Feature vector* yang akan dibuat berisi nilai dari tiga momen utama [pada masing-masing warna menggunakan *pseudocode* berikut:

```
Color_moment = zeros(1, 9);
Color_moment(1, :) = [meanR stdR
skR meanG stdG skG meanB stdB
skB];
```

Pada sintaks diatas himpunan bernama “Color_moment” dibuat dengan ukuran 9 kolom dalam 1 baris, menggunakan fungsi “zeros”. Kemudian variabel yang menampung nilai rata-rata, deviasi standar, dan kecenderungan dimasukkan pada himpunan tersebut. Hasil akhir dari proses ekstraksi fitur warna ini adalah data ciri berdasarkan warna dari citra yang telah dimasukkan. Data ciri warna tersebut disimpan pada himpunan *feature vector* yang bernama “Color_moment”. Gambar 3 memperlihatkan *feature vector* yang dihasilkan.

B. Fitur Tekstur

Tahapan proses ini adalah proses ekstraksi fitur kedua, di mana fitur yang akan diekstrak kali ini adalah fitur tekstur menggunakan filter Gabor[19]. Tahap awal melakukan ekstraksi citra kedalam *grayscale*, kemudian menjalankan filter gabor yang menghasilkan *feature vector* berisi nilai rata-

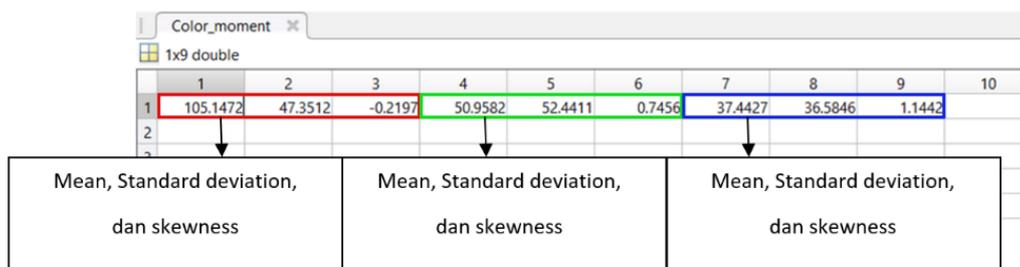
rata amplitudo, dan rata-rata energi [20]. Proses ekstraksi fitur tekstur menggunakan Filter Gabor perintah *pseudocode* Matlab berikut:

```
img = double(rgb2gray(image))/255;
[meanAmplitude, msEnergy] =
Gabor_filt(img);
```

Pada sintaks tersebut nilai citra yang ada pada “image” dibagi 255 untuk di konversi menjadi citra *grayscale* menggunakan fungsi “*rgb2gray*”. Setelah itu nilai hasil konversi disimpan pada variabel bernama “img” dengan menggunakan fungsi “*double*”. Hasil dari proses ini adalah data citra yang telah dikonversi menjadi citra *grayscale* dan siap digunakan untuk proses pada tahap selanjutnya. Tahapan proses yang dilakukan selanjutnya adalah proses menjalankan fungsi filter Gabor untuk mendapatkan nilai rata-rata

amplitudo, dan rata-rata energi. Pada sintaks diatas nilai citra *grayscale* yang ada pada variabel “img” diekstrak menggunakan fungsi “*Gabor_filt*”.

Setelah itu nilai hasil proses ekstraksi akan disimpan pada himpunan *feature vector* yang bernama “meanAmplitude” untuk nilai rata-rata amplitudo, dan “msEnergy” untuk nilai rata-rata energi. Hasil akhir dari proses ini adalah didapatkannya fitur atau ciri tekstur dari citra yang telah dimasukkan, dan fitur tersebut siap untuk digunakan pada tahap berikutnya. Gambar 4 merupakan contoh *feature vector* tekstur yang dihasilkan. Pada tahap ini, hasil ekstraksi fitur citra dari tahapan sebelumnya digabungkan menjadi satu feature vector, agar siap diproses pada pengukuran kemiripan fitur menggunakan perhitungan euclidean distance.



Gambar 3. Feature Vector Hasil Momen Warna

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	3.4032	8.8743	19.9843	46.2385	2.9687	7.2962	16.1646	34.4098	4.9049	11.6352	23.6879	54.8375	8.7602
2													
3													

Gambar 4. Feature Vector Hasil Filter Gabor

4. Penggabungan Dua Fitur

Tahapan proses ini adalah proses menggabungkan kedua fitur yang sudah di ekstrak pada tahap sebelumnya. Fitur warna dan tekstur yang telah di ekstrak di tahap sebelumnya masih disimpan di dalam variabel *feature vector* masing-masing ekstraktor seperti dapat dilihat pada Gambar 5. Pada Gambar 5, dapat dilihat *feature vector* “color_moment” dan “gabor_filter” dimasukkan ke dalam variabel himpunan baru bernama “ImageFeatures”. Variabel baru ini yang akan menampung data dua buah fitur yang sudah diekstrak pada tahap sebelumnya. Teknik yang dilakukan untuk proses penggabungan dua buah himpunan menggunakan *Pseudocode* berikut:

```
ImageFeatures = [color_moments  
meanAmplitude msEnergy];
```

Pada sintaks tersebut variabel bernama “ImageFeatures” menampung nilai dari 3 variabel *feature vector*. “color_moments” diletakan pada kolom pertama himpunan, “meanAmplitude” diletakan pada kolom kedua himpunan, dan “msEnergy” diletakan pada himpunan ketiga. Hasil dari proses ini adalah didapatkannya gabungan fitur atau ciri warna dan tekstur dari citra yang telah

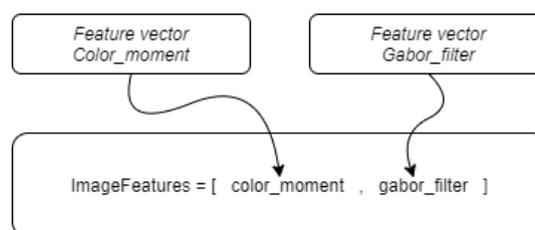
diekstrak pada tahap sebelumnya. Gambar 6 merupakan *feature vector* gabungan yang dihasilkan.

5. Pengukuran Kemiripan Fitur

Tahapan proses ini adalah proses pengukuran kemiripan fitur antara fitur dari citra permintaan atau *query* dengan kumpulan citra pada basis data. Metode yang digunakan untuk pengukuran kemiripan fitur pada penelitian ini menggunakan *euclidean distance* [21] menggunakan Persamaan :

$$\Delta E = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (1)$$

dimana: ΔE adalah Jarak *Euclidian* yang dihitung berdasarkan variabel (x_1, y_1) sebagai jarak titik awal dan (x_2, y_2) yang merupakan jarak titik berikutnya. Semakin kecil atau mendekati nol nilai jarak euclidean antara kedua buah citra maka akan semakin mirip kedua citra tersebut. Apabila nilai jarak euclidean adalah nol antara kedua buah citra maka kedua citra tersebut adalah citra yang sama. Fitur dari kumpulan citra pada basis data disimpan di file yang bernama “dataset.mat”. Fitur dari citra permintaan atau *query* disimpan pada variabel bernama “queryImageFeature”.



Gambar 5. Penggabungan Dua Fitur

Feature Vector Color Moment									
1	2	3	4	5	6	7	8	9	10
105.1472	47.3512	-0.2197	50.9582	52.4411	0.7456	37.4427	36.5846	1.1442	
2									
3									

Feature Vector Gabor Filter								
1	2	3	4	5	6	7	8	9
3.4032	8.8743	19.9843	46.2385	2.9687	7.2962	16.1646	34.6098	4.9049
2								
3								

Gambar 6. Hasil Penggabungan 2 Feature Vector



Gambar 7. Citra Satu dan Dua

Setelah fitur kedua citra siap di proses maka fitur dihitung menggunakan persamaan *euclidean distance* menggunakan *pseudocode* berikut :

Euclidean = sqrt(sum(power(dataset-queryImageFeatureVector, 2)));

Pada sintaks diatas nilai dari fitur citra pada dataset yang dikurang dengan fitur citra *query*, hasil dari pengurangan tersebut dipangkat dua lalu dijumlahkan dengan fungsi “sum”, setelah itu hasil diakarkan dengan fungsi “sqrt” untuk mendapat nilai jarak euclidean yang ditampung pada variabel “Euclidean”. Contoh dari perhitungan jarak euclidean antara dua buah citra pada Gambar 7 dapat dilihat sebagai berikut:

Vektor fitur citra satu = [4, 1, 2, 6]

Vektor fitur citra dua = [3, 2, 1, 3]

Euclidean

$$\begin{aligned}
 &= \sqrt{(4 - 3)^2 + (1 - 2)^2 + (2 - 1)^2 + (6 - 3)^2} \\
 &= \sqrt{(1)^2 + (1)^2 + (1)^2 + (3)^2} \\
 &= \sqrt{1 + 1 + 1 + 9} \\
 &= \sqrt{12} \\
 &= 3.46410162
 \end{aligned}$$

Penggunaan persamaan *euclidean distance* maka didapatkan nilai jarak antara citra satu dan citra dua adalah 3,46410162. Nilai jarak yang dihasilkan akan diurutkan dari nilai terkecil hingga terbesar, lalu akan diambil 10 citra dengan nilai jarak terkecil untuk ditampilkan. Pada Tabel 2 dapat dilihat contoh 10 citra yang diambil berdasarkan nilai jarak yang dihasilkan.

Tabel 2. Contoh 10 Citra dengan Nilai Jarak Terkecil

<i>Citra Query</i>	No	<i>Citra Hasil</i>	Nilai <i>Euclidean Distance</i>
	1		0
	2		83.5061
	3		86.4742
	4		88.5404
	5		88.5479
	6		89.6132
	7		89.6665
	8		90.4387
	9		91.6116
	10		91.6215

HASIL DAN PEMBAHASAN

Pada penelitian ini, tahap ekstraksi fitur diharapkan menghasilkan vektor fitur atau

feature vector dari masing-masing ekstraktor. Setelah kedua *feature vector* didapatkan, maka *feature vector* digabungkan ke satu variabel agar siap digunakan pada tahap berikutnya.

Pada tahap ekstraksi fitur ini citra yang digunakan dapat dilihat pada Gambar 8.

A. Hasil Ekstraksi Fitur Warna

Hasil dari proses ekstraksi fitur warna ini dilakukan dengan citra pada Gambar 8. Ekstraktor yang digunakan adalah *color moment*, di mana hasil ekstraksinya akan menghasilkan *feature vector* bernama *Color_moment*. Gambar 9 merupakan hasil vektor fitur Momen Warna yang dihasilkan dari ekstraksi pada citra Gambar 8. Terdapat keterangan yang menjelaskan untuk kolom 1 sampai dengan 3 merupakan nilai *mean*,

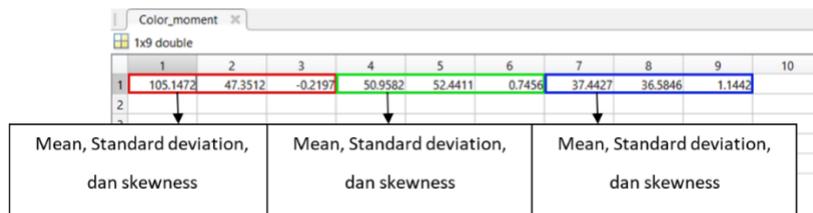
standard deviation, dan *skewness* dari saluran warna merah, kolom 4 sampai dengan 6 untuk saluran warna hijau, dan kolom 7 sampai dengan 9 untuk saluran warna biru, sehingga dihasilkan vektor berukuran 1x9.

B. Hasil Ekstraksi Fitur Tekstur

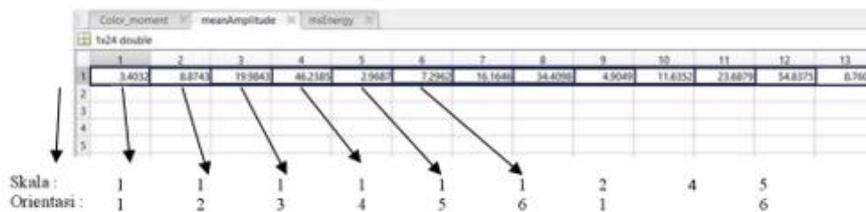
Hasil dari proses ekstraksi fitur tekstur ini dilakukan dengan citra pada Gambar 8. Ekstraktor yang digunakan adalah Filter Gabor, di mana hasil ekstraksinya akan menghasilkan 2 *feature vector* yaitu *meanAmplitude* dan *msEnergy* seperti dapat dilihat pada Gambar 10.



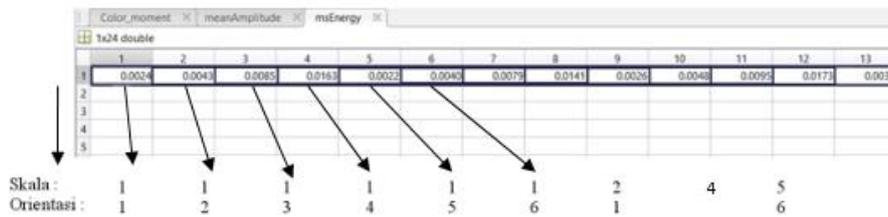
Gambar 8. Citra yang digunakan



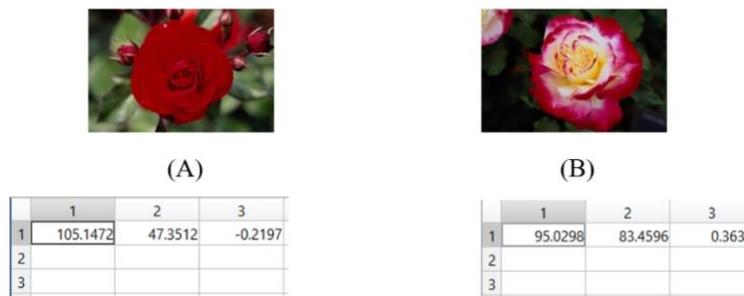
Gambar 9. Feature Vector Hasil Momen Warna Citra Gambar 8



Gambar 10. Feature Vector meanAmplitude Hasil Gabor Filter



Gambar 11. Feature Vector msEnergy Hasil Gabor Filter



Gambar 12. Citra Kategori yang Sama dengan Feature Vector yang Berbeda

Gambar 10 merupakan vektor fitur meanAmplitude dari ekstraktor *Gabor Filter*, yang didapatkan dari ekstraksi pada citra 8. Terdapat keterangan untuk setiap kolom berisi nilai meanAmplitude citra pada setiap skala dan rotasi. Pada ekstraktor filter gabor yang digunakan di penelitian ini, skala yang digunakan adalah 4 dan orientasi yang digunakan adalah 6 [22], sehingga dihasilkan vektor berukuran 1x24 seperti dapat dilihat pada Gambar 10. Gambar 11 merupakan hasil *feature vector* atau vektor fitur msEnergy dari ekstraktor *Gabor Filter*, yang didapatkan dari ekstraksi pada citra Gambar 8. Terdapat keterangan untuk setiap kolom berisi nilai msEnergy citra pada setiap skala dan rotasi. Pada ekstraktor filter gabor yang digunakan di penelitian ini, skala yang digunakan adalah 4

dan orientasi yang digunakan adalah 6 [22], sehingga dihasilkan vektor berukuran 1x24.

C. Hasil Pengukuran Kemiripan Fitur

Pada penelitian ini, tahap pengukuran kemiripan fitur akan dilakukan pada citra dikategori bunga pada Gambar 8. Citra tersebut akan diukur dengan persamaan *euclidean distance*. Pengukuran citra pada Gambar 8 akan dilakukan dengan citra itu sendiri, dengan citra dikategori yang sama, dan dengan citra dikategori yang berbeda.

1. Hasil dengan Citra Kategori yang Sama

Pada tahap pengukuran kemiripan fitur ini, citra yang akan diukur dengan citra pada dengan citra dikategori yang sama yaitu kategori bunga seperti dapat dilihat pada Gambar 12.

Seperti dapat dilihat pada Gambar 12 terdapat 2 citra yang masuk kategori yang sama yaitu “bunga” akan tetapi memiliki nilai feature vector yang berbeda. Sebagai contoh pada nilai *Mean*, *Standard Deviation*, dan *Skewness* untuk citra A memiliki nilai 105,1472 dan pada citra B memiliki nilai 95,0298. Fitur citra yang dihitung adalah fitur citra pada Gambar A dengan citra pada Gambar B menghasilkan nilai *euclidian distance* 61,7058 dari proses perhitungan. Nilai 61,7058 tersebut dapat diartikan kedua citra yang diukur adalah bukan citra yang sama.

2. Hasil dengan Citra Kategori Berbeda

Pada tahap pengukuran kemiripan fitur ini, citra yang akan diukur dengan citra dikategori yang berbeda dengan yaitu kategori kuda seperti dapat dilihat pada Gambar 13. Hasil dari proses perhitungan *euclidean distance* menghasilkan nilai 192,4789 dari proses perhitungan. Nilai 192,4789 tersebut dapat diartikan kedua citra yang diukur adalah

bukan citra yang sama. Tetapi jika dibandingkan dengan nilai pada citra kategori yang sama, nilai kategori yang sama lebih kecil dibandingkan dengan nilai pada kategori yang berbeda.

D. Hasil Ujicoba Presisi

Pada penelitian ini, uji coba presisi akan dilakukan untuk menguji presisi dari setiap kategori. Data uji yang digunakan adalah sepuluh citra yang diambil secara acak dari basis data. Sepuluh citra yang diambil tersebut mewakili setiap kategori sebagai data uji. Pengujian subjektif juga dilakukan untuk mengetahui apakah citra hasil yang dikeluarkan relevan atau tidak. Setelah ke sepuluh kategori selesai diuji, maka ke sepuluh nilai presisi yang didapatkan dari setiap kategori akan dihitung rata-ratanya. Sebagai contoh ditampilkan 2 dari hasil uji berdasarkan hasil dari presisi terendah presisi pada kategori gunung dan presisi paling tinggi sejumlah pada kategori dinosaurus.



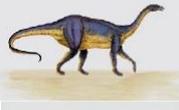
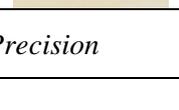
Gambar 13. Citra Kategori yang Berbeda

1. Kategori Dinosaurus

Pada Tabel 3 dapat dilihat citra *query* yang dimasukkan, citra hasil yang ditampilkan, nilai euclidean distance dari citra hasil, pengujian subjektif, dan nilai presisi yang didapatkan pada kategori dinosaurus. Hasil

pengujian pada data uji kategori dinosaurus didapatkan 10 citra relevan dengan kategori dan tidak ada citra yang tidak relevan dengan kategori. Sehingga didapatkan presisi kategori dinosaurus adalah 10/10 dengan persentase 100%.

Tabel 3. Hasil Uji Presisi Kategori Dinosaurus

Citra <i>Query</i>	No	Citra Hasil	Nilai <i>Euclidean Distance</i>	Pengujian Subjektif
	1		0	Benar
	2		40.7992	Benar
	3		57.4064	Benar
	4		58.8549	Benar
	5		61.7207	Benar
	6		63.2158	Benar
	7		63.5488	Benar
	8		63.5821	Benar
	9		64.8172	Benar
	10		67.5871	Benar
<i>Precision</i>				10/10 = 1 x 100% = 100%

Tabel Error! No text of specified style in document.. Hasil Uji Presisi Kategori Gunung

Citra Query	No	Citra Hasil	Nilai <i>Euclidean Distance</i>	Pengujian Subjektif
	1		0	Benar
	2		83.5061	Salah
	3		86.4742	Benar
	4		88.5404	Salah
	5		88.5479	Benar
	6		89.6132	Salah
	7		89.6665	Salah
	8		90.4387	Benar
	9		91.6116	Salah
	10		91.6215	Benar
<i>Precision</i>			$5/10 = 0.5 \times 100\%$ $= 50\%$	

2. Kategori Gunung

Pada Tabel 4 dapat dilihat citra *query* yang dimasukkan, citra hasil yang ditampilkan, nilai euclidean distance dari citra hasil, pengujian subjektif, dan nilai presisi yang didapatkan pada kategori gunung. Hasil

pengujian pada data uji kategori gunung didapatkan 5 citra relevan dengan kategori dan 5 citra tidak relevan dengan kategori. Sehingga didapatkan presisi kategori gunung adalah 5/10 dengan persentase 50%.

KESIMPULAN DAN SARAN

Berdasarkan hasil ujicoba presisi sistem Temu Kenali Citra Berbasis Konten berdasarkan warna dan tekstur yang telah dibuat pada penelitian ini, dapat diambil beberapa kesimpulan antara lain: Sistem Temu Kenali Citra Berbasis Konten dapat mengekstrak citra dengan ekstraktor Momen Warna dan Filter Gabor untuk menghasilkan *feature vector* warna dan tekstur. Penelitian ini berhasil menemukan 10 citra yang dicari berdasarkan nilai dari perhitungan *euclidean distance* pada *feature vector* citra *query* dengan citra pada basis data dengan persentase presisi tingkat keberhasilan sistem yang diuji pada setiap kategori menggunakan pengujian secara visual dengan memperhatikan citra *groundtruth*. Hasil paling rendah memiliki presisi sejumlah 50% pada kategori gunung dan presisi paling tinggi sejumlah 100% pada kategori dinosaurus. Rata-rata persentase presisi tingkat keberhasilan sistem menggunakan pengujian subjektif, didapatkan hasil 84% dari 10 data uji yang diambil dari database. Berdasarkan hasil penelitian yang dilakukan, dapat diberikan beberapa saran antara lain: Penambahan jumlah dataset, penggunaan fitur citra lain seperti tekstur dan bentuk, dan penambahan metode pengukuran kemiripan fitur lain untuk mendapatkan citra *query* yang sesuai dengan citra *groundtruth*.

DAFTAR PUSTAKA

- [1] M. K. Ahirwal, A. Kumar, and G. K. Singh, "An approach to design self assisted CBIR system", Proc. Int. Conf. Graph. Signal Process. – ICGSP, Vol. 17, pp.21–25, 2017.
- [2] K. T. Ahmed, S. Ummesafi, and A. Iqbal, "Content based image retrieval using image features information fusion", Information Fusion, vol. 51, pp.76–99, 2019.
- [3] M. K. Alsmadi, "An efficient similarity measure for content based image retrieval using memetic algorithm", Egyptian Journal of Basic and Applied Sciences, vol. 4, no. 2, pp.112–122, 2017.
- [4] M. K. Alsmadi, "Query-sensitive similarity measure for content-based image retrieval using meta-heuristic algorithm", Journal of King Saud University-Computer and Inf. Sci., vol. 30, no. 3, pp.373–381, 2018.
- [5] M. Bouchakwa, Y. Ayadi, and I. Amous "Multi-level diversification approach of semantic-based image retrieval results", Progress in Artificial Intelligence, vol. 9, no. 1, pp.1–30, 2020.
- [6] A. Du, L. Wang, and J. Qin, "Image retrieval based on colour and improved NMI texture features", Automatika, vol. 60, no. 4, pp.491–499, 2019.
- [7] M. A. Aziz, A. A. Ewees, and A. E. Hassanien, "Multi-objective whale

- optimization algorithm for content-based image retrieval”, *Multimed. Tools Appl.*, vol. 77, no. 19, pp.26135–26172, 2018.
- [8] D. A. Makandar, R. Somshekhar, and N. Jaday, “Content based image retrieval”, *World Applied Sciences Journal*, 19(3), 404–412. <https://doi.org/10.5829/idosi.wasj.2012.19.03.1506>, 2019.
- [9] R. Rajkumar, and M. V. Sudhamani, “Content based Image Retrieval System using Combination of Color and Shape Features, and Siamese Neural Network”, *International Journal of Innovative Technology and Exploring Engineering*, 9(2S), 71–77. <https://doi.org/10.35940/ijitee.b1053.1292s19>, 2019.
- [10] T. N. Phalke, and A. Patil, “Content Based Image Retrieval Using Color And Texture”, 9(1), 992–1000. <https://doi.org/10.5121/sipij.2012.3104>, 2017.
- [11] N. Varish, and A. K. Pal, “A content based image retrieval using color and texture features”, *ACM International Conference Proceeding Series*, 12-13. <https://doi.org/10.1145/2979779.2979787>, 2016.
- [12] B. Babu, R. Vanitha, and K. S. Anish, “Content based image retrieval using color, texture, shape and active re-ranking method”, *Indian Journal of Science and Technology*, 9(17). <https://doi.org/10.17485/ijst/2016/v9i17/93107>, 2016.
- [13] J. Li, and J. Z. Wang, “Automatic linguistic indexing of pictures by a statistical modeling approach”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 9, pp. 1075-1088, 2003.
- [14] A. Amelio, “A new axiomatic methodology for the image similarity”, *Applied Soft Computing*, vol. 81, p. 105474, 2019
- [15] Z. Yu, and W. Wang, “Learning DALTS for cross-modal retrieval,” *CAAI Transactions on Intelligence Technology*, vol. 4, no. 1, pp. 9–16, 2019.
- [16] B. Zafar, R. Ashraf, N. Ali, and M. Ahmed., “Intelligent image classification-based on spatial weighted histograms of concentric circles”, *Computer Science and Information Systems*, vol. 15, no. 3, pp. 615–633, 2018.
- [17] S. Fadaei, R. Amirfattahi, and M. R. Ahmadzadeh, “Local derivative radial patterns: a new texture descriptor for content-based image retrieval”, *Signal Processing*, vol. 137, pp. 274–286, 2017.
- [18] A. Singla, M. Garg “CBIR Approach Based On Combined HSV, Auto Correlogram, Color Moments And Gabor Wavelet”, *International Journal Of Engineering And Computer Science*,

- ISSN:2319-7242, vol. 3, pp. 9007–9012, Oct, 2014.
- [19] M. Kuse, “Filter Gabor Library”, sumber:
<https://www.mathworks.com/matlabcentral/fileexchange/38844-gabor-image-features>, 1 Februari 2021.
- [20] P. Pakutharivu, and M. V. Srinath, "Analysis of Fingerprint Image Enhancement Using Gabor Filtering with Different Orientation Field Values", Indonesian Journal of Electrical Engineering and Computer Science, vol. 5, pp. 427- 432, Feb, 2017.
- [21] D. T. Susetianingtias, H.S. Suryadi, S. Madenda, Rodiah, and Fitriyaningsih, "Blood vessel extraction and bifurcations detection using hessian matrix of gaussian and euclidian distance", Journal of Theoretical and Applied Information Technology 95(15):3471-3478, Aug, 2017.
- [22] M. Kuse, V. Kalasannavar, N. Rajpoot, Y. Wang, and M. Khan, “Local isotropic phase symmetry measure for detection of beta cells and lymphocytes”, Journal of Pathology Informatics, 2(2), 2. <https://doi.org/10.4103/2153-3539.92028>, 2011.