

Content-Based Image Retrieval berbasis Warna, Bentuk dan Tekstur dengan Modifikasi Skala

Arwin Halim¹, Alex Xandra Albert Sim², Hardy³, Mytosin⁴

STMIK Mikroskil, Jl. Thamrin No. 112, 124, 140, Telp. (061) 4573767, Fax. (061) 4567789

^{1,2,3,4}Jurusan Teknik Informatika, STMIK Mikroskil, Medan

¹arwin@mikroskil.ac.id, ²alex.sim@mikroskil.ac.id, ³hardy@mikroskil.ac.id,

⁴mytosin.liu@mikroskil.ac.id

Abstrak

Pencarian gambar dapat dilakukan berdasarkan informasi fitur dari gambar. Pada penelitian ini, kombinasi fitur gambar (warna, tekstur dan bentuk) diterapkan dalam pendekatan yang berbeda, yaitu adanya skala pada gambar. Setiap gambar dengan skala yang berbeda melalui tahapan smoothing. Tahapan ini berfungsi untuk memperoleh nilai fitur yang lebih representatif dan mencegah pendeteksian piksel gambar yang tidak diperlukan. Dataset yang digunakan berasal dari dataset Wang. Tahapan berikutnya adalah perhitungan fitur gambar berdasarkan fitur warna dengan color histogram, fitur tekstur dengan Gray-Level Co-occurrence Matrix, dan fitur bentuk dengan polar raster edge sampling signature. Hasil pencarian dihitung dari nilai Euclidean distance. Nilai f-measure digunakan untuk mengevaluasi hasil pencarian berdasarkan nilai presisi dan recall. Hasil penelitian menunjukkan kombinasi fitur warna, tekstur dan bentuk dapat meningkatkan keakuratan hasil pencarian gambar. Selain itu, perubahan skala gambar mampu meningkatkan hasil pencarian gambar berdasarkan fitur bentuk.

Kata kunci— content-based image retrieval, fitur bentuk, fitur tekstur, ruang warna, skala gambar

Abstract

Image retrieval can be performed based on the features information from the image. In this study, the combination of image features (color, texture, and shape) is applied in the different approach that is the image scaling. Every images with different scales were processed at smoothing stage. This process serves to obtain a representative number of features and prevent the detection of image pixels that are not necessary. The dataset comes from the dataset Wang. The next stage is the computation of image features based on color features with color histogram, texture features with Gray-Level Co-occurrence Matrix, and texture features with polar raster edge sampling signature. The results calculated from the Euclidean Distance. F-Measure is used to evaluate the results based on the precision and recall values. The results showed the combination of features of colors, textures and shapes can increase the accuracy of the image retrieval. In addition, changes in the image scale can improve accuracy of image retrieval based on shape features.

Keywords— color space, content-based image retrieval, image scaling, shape feature, texture feature

1. PENDAHULUAN

Perkembangan informasi multimedia gambar dan teknologi fotografi menyebabkan jumlah gambar digital bertambah dengan cepat. Pencarian gambar menjadi semakin rumit karena mengandung banyak objek yang kompleks dan memiliki rincian informasi warna yang beragam [1]. Pencarian gambar dapat dilakukan dengan pendekatan *text-based* dan *content-based*. Pendekatan *text-based* sangat bergantung pada interaksi manusia, seperti kata kunci, judul gambar, deskripsi dan lain-lain, sehingga memiliki kesulitan dan keterbatasan dalam menggambarkan isi gambar pada teks. Pendekatan lain dalam pencarian gambar adalah *content-based* yang mencoba mengolah informasi dari fitur gambar

yang dicari, seperti warna, bentuk, tekstur, atau informasi lainnya [2]. Pada umumnya, pendekatan *content-based* menggunakan gambar asli untuk mengekstrak fitur-fiturnya. Perubahan skala pada gambar tidak secara langsung diperhitungkan dalam pencarian gambar.

Informasi dari fitur gambar dapat berupa fitur warna, bentuk dan tekstur. Penelitian Ali dkk [3] telah menggunakan fitur warna dengan color space RGB dan HSV untuk mendeteksi gambar tanda jalan. Hasil penelitian tersebut menunjukkan, HSV memberikan hasil yang lebih baik dibandingkan dengan RGB dalam pendeteksian gambar tanda jalan. Penelitian Arthi dan Vijayaraghavan [4] menggunakan fitur warna yaitu Colour Co-occurrence Matrix dengan model HSV dalam *content-based image retrieval*. Pendekatan fitur warna lain yang populer digunakan adalah color histogram. Selain fitur warna, pencarian gambar dengan kombinasi fitur juga telah dilakukan. Penulis telah menggabungkan fitur warna & tekstur [5] dan fitur warna & bentuk [6]. Selain itu, penelitian lain juga ada yang menggabungkan metode Color-Histogram untuk fitur warna dan Gray-Level Co-occurrence Matrix (GLCM) untuk fitur tekstur dapat digunakan untuk mendapatkan hasil pencarian yang baik [7]. Penelitian lainnya yang memperhitungkan perbedaan ukuran gambar adalah Multi-scale GLCM oleh Siqueira dkk [8]. Selain warna dan tekstur, pendeteksian gambar berdasarkan bentuk menggunakan deteksi tepi telah dikembangkan. Penulis telah menggunakan deteksi tepi Sobel sebagai salah satu fitur dalam pencarian gambar [9]. Algoritma deteksi tepi lainnya untuk melakukan pencarian gambar pada database besar adalah Polar raster edge sampling signature [10].

Permasalahan pada penelitian ini adalah perlunya melakukan kajian pengaruh perubahan skala gambar dalam pencarian gambar pada database berdasarkan fitur warna, bentuk dan tekstur. Beberapa hal yang menjadi fokus antara lain:

1. Data penelitian yang digunakan berasal dari dataset standar Wang dengan karakteristik:
 - a. Gambar memiliki format jpg
 - b. Ukuran gambar pada dataset adalah 384x256
 - c. 10 kategori gambar yaitu afrika, pantai, bangunan, bus, dinosaurus, gajah, bunga, kuda, gunung dan makanan.
2. Teknik smoothing yang digunakan dalam *preprocessing* adalah Gaussian-smoothing
3. Fitur warna menggunakan algoritma Histogram Warna (CH)
4. Fitur tekstur diperoleh dari perhitungan Gray-Level Co-occurrence Matrix (MSGLCM)
5. Fitur bentuk diperoleh dari perhitungan deteksi tepi menggunakan algoritma Polar Raster Edge Sampling Signature (PRESS).

2. METODE PENELITIAN

2.1. Kajian Pustaka

2.1.1. Histogram Warna

Histogram warna [11] adalah representasi distribusi warna pada gambar. Untuk gambar digital, histogram warna mewakili jumlah *pixel* yang memiliki warna pada masing-masing kelompok, dengan rentang warna tertentu yang mencakup ruang warna gambar tersebut. Langkah-langkah untuk menghitung fitur warna, yaitu:

- a. Konversi ruang warna

Pada tahap ini, ubah semua warna pada gambar dari ruang warna RGB (*Red, Green, Blue*) menjadi ruang warna HSV. HSV adalah *Hue, Saturation, Value*. *Hue* digunakan untuk mewakili warna, *saturation* adalah banyaknya cahaya putih yang ditambahkan ke warna dasar, dan *value* adalah intensitas cahaya. Perhitungan nilai HSV diperoleh dari Persamaan 1 dan 2.

$$\begin{aligned}
 R' &= \frac{R}{255} & G' &= \frac{G}{255} & B' &= \frac{B}{255} \\
 C_{max} &= \max(R', G', B') \\
 C_{min} &= \min(R', G', B') \\
 \Delta &= C_{max} - C_{min}
 \end{aligned}$$

(1)

$$\begin{aligned}
 H &= \begin{cases} 60^\circ \times \left(\frac{G' - B'}{\Delta} \bmod 6 \right), C_{max} = R' \\ 60^\circ \times \left(\frac{B' - R'}{\Delta} + 2 \right), C_{max} = G' \\ 60^\circ \times \left(\frac{R' - G'}{\Delta} + 4 \right), C_{max} = B' \end{cases} \\
 S &= \begin{cases} 0, C_{max} = 0 \\ \frac{\Delta}{C_{max}}, C_{max} \neq 0 \end{cases} \\
 V &= C_{max}
 \end{aligned} \tag{2}$$

b. Kuantisasi nilai HSV

Pada ruang warna HSV, setiap komponen memiliki rentang nilai yang besar. Jika menggunakan nilai H, S, dan V secara langsung, maka memerlukan banyak komputasi. Jadi lebih baik jika ruang warna HSV dikuantisasi secara *non-equal intervals*, yaitu H: 8 bagian, S: 3 bagian, dan V: 3 bagian, yang menggunakan Persamaan 3.

$$\begin{aligned}
 H &= \begin{cases} 0 \text{ if } h \in [316, 20] \\ 1 \text{ if } h \in [21, 40] \\ 2 \text{ if } h \in [41, 75] \\ 3 \text{ if } h \in [76, 155] \\ 4 \text{ if } h \in [156, 190] \\ 5 \text{ if } h \in [191, 270] \\ 6 \text{ if } h \in [271, 295] \\ 7 \text{ if } h \in [296, 315] \end{cases} \\
 S &= \begin{cases} 0 \text{ if } s \in [0, 0.2] \\ 1 \text{ if } s \in [0.2, 0.7] \\ 2 \text{ if } s \in [0.7, 1] \end{cases} \\
 V &= \begin{cases} 0 \text{ if } v \in [0, 0.2] \\ 1 \text{ if } v \in [0.2, 0.7] \\ 2 \text{ if } v \in [0.7, 1] \end{cases}
 \end{aligned} \tag{3}$$

Kemudian digabungkan menjadi 8x3x3 histogram dan mendapatkan vektor 72 dimensi. Karena kemampuan mata manusia untuk membedakan warna sangat terbatas, vektor 3 dimensi untuk kombinasi nilai H, S, dan V akan dinormalisasi menjadi vektor 1 dimensi dengan Persamaan 4. Ini akan mengkuantisasi seluruh ruang warna HSV menjadi 72 bagian warna dengan 1 dimensi.

$$G = 9H + 3S + V \tag{4}$$

c. Hitung Histogram. Pada tahap ini, gambar yang sudah dikuantisasi akan dihitung histogram warnanya, yaitu frekuensi distribusi nilai HSV setiap *pixel* pada gambar yang telah dikuantisasi.

2.1.2. Gray-Level Co-occurrence Matrix (GLCM)

GLCM dapat memberikan informasi tentang distribusi spasial dari *gray-level* pada tekstur gambar [8]. GLCM menandakan seberapa sering sebuah *pixel* dengan nilai *i* berhubungan secara spasial dengan *pixel* yang bernilai *j*. Pada setiap GLCM, masing-masing elemen $p(i, j)$ adalah total kemunculan suatu *pixel* *i* dengan *pixel* *j* untuk arah tertentu (0, 45, 90, 135). Untuk menghitung fitur akhir GLCM, maka hitung fitur untuk empat arah GLCM dan hitung rata-rata fiturnya. Pada dasarnya, GLCM adalah sebuah *matrix* simetris dan levelnya ditentukan oleh *gray-level* dari gambar. Intensitas nilai *grayscale* pada gambar dikurangi dari 256 menjadi 8, sehingga *matrix*-nya berukuran 8x8. Elemen pada *matrix* dihitung dengan Persamaan 5.

$$P(i, j | d, \theta) = \frac{P(i, j | d, \theta)}{\sum_i \sum_j P(i, j | d, \theta)} \tag{5}$$

Metode GLCM menghasilkan empat fitur gambar antara lain:

a. *Energy*

Energy adalah pengukuran tekstur dari gambar *grayscale* yang merepresentasikan perubahan *homogeneity*, merefleksikan keseragaman berat dan tekstur distribusi gambar *grayscale*, yang menggunakan Persamaan 6.

$$E = \sum_x \sum_v P(x, y)^2 \quad (6)$$

b. *Contrast*

Contrast adalah diagonal utama yang mengukur nilai *matrix* yang terdistribusi, merefleksikan kejelasan gambar dan tekstur kedalaman bayangan, yang menggunakan Persamaan 7.

$$I = \sum \sum (x - y)^2 P(x, y) \quad (7)$$

c. *Correlation*

Correlation mengukur keteraturan tekstur gambar, ketika setiap nilai *matrix* sama besar, maka menghasilkan nilai korelasi yang minimum, dan sebaliknya jika nilainya sangat tidak merata, nilai korelasinya akan semakin besar. *Correlation* dihitung dengan menggunakan Persamaan 8.

$$C = - \sum_x \sum_y P(x, y) \log P(x, y) \quad (8)$$

d. *Homogeneity*

Homogeneity mengukur nilai perubahan lokal pada tekstur gambar, dengan Persamaan 9.

$$H = \sum_x \sum_y \frac{1}{1+(x-y)^2} P(x, y) \quad (9)$$

2.1.3. Polar-Raster Edge Sampling Signature

Polar-Raster Edge Sampling Signature (PRESS) [10] adalah teknik untuk mengekstrak fitur bentuk dari gambar. Secara garis besar, tahapan PRESS terdiri dari segmentasi gambar, ekstraksi fitur dan pencarian gambar. Segmentasi gambar dilakukan pada gambar berdasarkan warna gambar menggunakan algoritma K-Means clustering. Gambar tersebut kemudian diproses kembali dengan menggunakan algoritma Canny untuk mendeteksi setiap tepi pada gambar. Strong edge dan connected edge diidentifikasi dengan menggunakan teknik double thresholding dan edge tracking. Nilai tepi yang diperoleh dan areanya akan di-scan dan dipetakan ke dalam radius dan sudut. Jumlah poin edge yang teridentifikasi akan disimpan sebagai nilai fitur pada gambar.

Pseudocode untuk Polar Raster Edge Sampling Signature:

1. Ubah gambar menjadi grayscale
2. Lakukan segmentasi gambar dengan menggunakan algoritma K-means
3. Temukan tepi gambar dengan menggunakan algoritma Canny
4. Lakukan smoothing pada gambar
5. Hitung polar transform yang menyimpan posisi piksel tepi pada gambar
6. Simpan nilai radius dan sudutnya sebagai fitur

```
BW ← edge(mask, 'canny'); // Finding the edges of the image.
```

```
[imx, imy] ← size(BW);
```

```
B ← conv2(BW, msk); //Smoothing the image
```

```
[x,y] ← find(B == 1);
```

```
p = [x, y];
```

```
[qr, qt] = polartransform(p);
```

```
get count of edge points;
```

```
store radius bin count in vector r;
```

```
store angle bin count in vector t; //PRESS is used to extract shape features
```

```
normalize for sum of counts;
```

```
repeat for all images in the DB;
```

```
savethem in feature DB;
```

2.1.3. Euclidean Distance

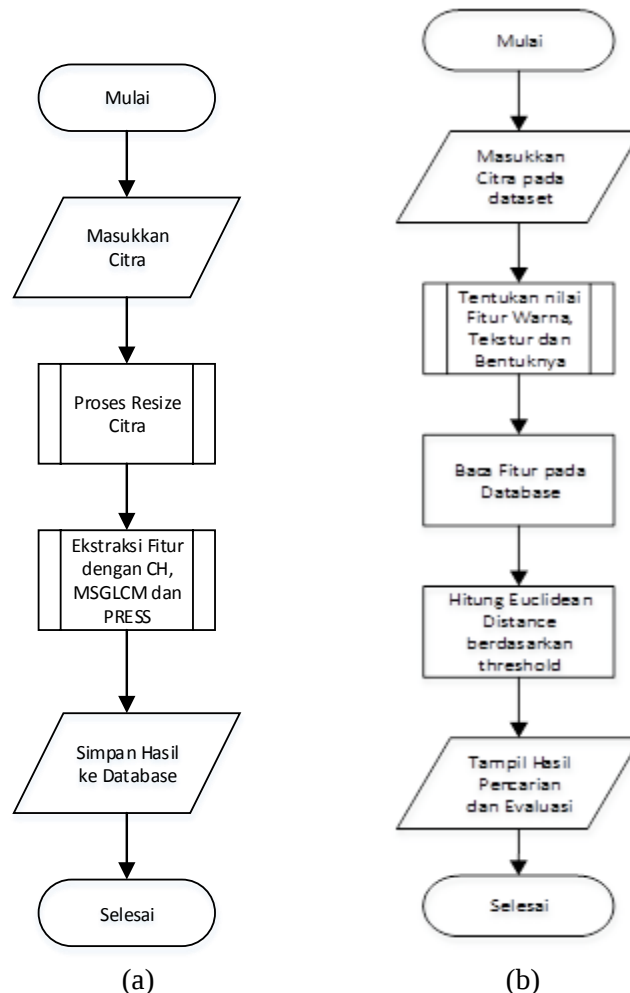
Euclidean distance [2] mengukur tingkat kemiripan antara dua fitur yang berbeda. Rumus untuk mengukur euclidean distance dapat dilihat pada Persamaan 10. Nilai Q menunjukkan fitur untuk Query image dan nilai D menunjukkan fitur untuk Database image.

$$d(Q, D) = \sqrt{\sum_{i=1}^n [Q_i - D_i]^2} \quad (10)$$

2.2. Analisis

2.2.1 Analisis Proses

Proses pencarian gambar terdiri dari proses *training* dan *testing*. Proses *training* gambar dilakukan untuk seluruh gambar dengan skala yang berbeda (50%, 60%, 70%, 80% dan 90% dari gambar aslinya). Proses *training* menghasilkan fitur warna, tekstur dan bentuk dari gambar yang di-train. Rincian proses *training* dapat dilihat pada Gambar 1(a). Setelah data di-train, maka aplikasi telah dapat menemukan gambar yang memiliki tingkat kemiripan yang sesuai dengan threshold melalui proses *testing*. Rincian proses *testing* dapat dilihat pada Gambar 1(b).



Gambar 1 Proses Training (a) dan Testing (b) pada Aplikasi CBIR

Hasil pencarian gambar akan ditampilkan dengan urutan jarak terkecil sampai jarak terbesar dan tidak melebihi nilai threshold yang ditentukan. Hasil pencarian digunakan untuk menghitung nilai presisi dan recall untuk mendapatkan keakuratan dari pencarian. Nilai presisi dan recall dapat dihitung dengan Persamaan 11.

$$\begin{aligned}
 \text{Precision} &= \frac{\text{retrieved relevant images}}{\text{retrieved images}} \\
 \text{Recall} &= \frac{\text{retrieved relevant images}}{\text{relevant images in collection}}
 \end{aligned}
 \tag{11}$$

2.2.2 Analisis Hasil

Tahapan ini akan menguji kemampuan dari kombinasi algoritma color histogram, multiscale GLCM dan PRESS dalam melakukan pencarian gambar yang memiliki keakuratan yang lebih baik. Pengujian yang dilakukan berupa pencarian untuk setiap gambar pada dataset (Leave One Out Cross Validation). Hasil pencarian dievaluasi dengan menghitung nilai presisi dan recall. Evaluasi dilakukan lebih lanjut dengan menghitung nilai F-measure. Nilai F-measure dapat dihitung berdasarkan Persamaan 12.

$$fmeasure = 2 * \frac{precision * recall}{precision + recall} \tag{12}$$

Nilai F-measure yang tinggi menunjukkan pengujian akurasi yang lebih baik. Nilai maksimum dari F-measure adalah 1 yang berarti semua hasil pencarian benar dan ditampilkan semua. Nilai F-measure = 0 berarti semua hasil pencarian salah.

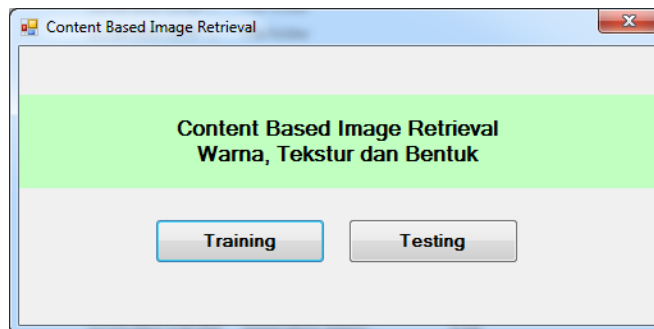
3. HASIL DAN PEMBAHASAN

3.1 Hasil

Penelitian ini menghasilkan aplikasi pencarian gambar berbasis desktop yang dibangun dengan bahasa pemrograman C# .NET Framework 4.5. Tampilan dari aplikasi dapat dijelaskan sebagai berikut:

1. Tampilan awal

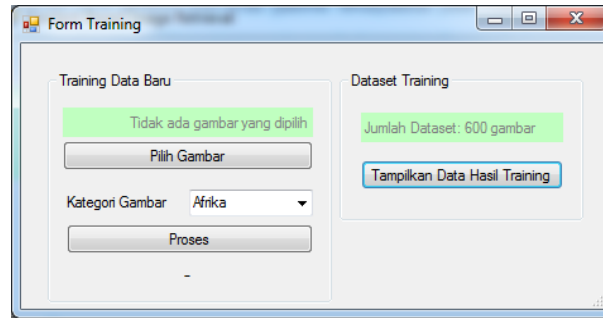
Tampilan awal aplikasi pencarian gambar ditunjukkan pada Gambar 2. Tampilan awal ini terdiri dari dua pilihan utama yaitu tombol *training* dan tombol *testing*.



Gambar 2 Tampilan Awal Aplikasi Pencarian Gambar

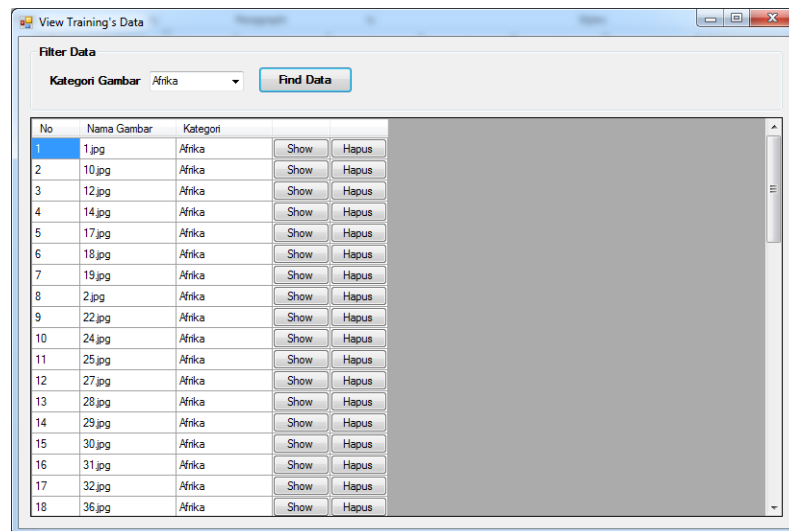
2. Tampilan *training*.

Tampilan *training* akan muncul ketika pengguna memilih tombol *training* pada tampilan awal aplikasi. Tampilan *training* ditunjukkan pada Gambar 3. Tampilan *training* terdiri dari dua bagian utama yaitu bagian untuk *training* data baru dan bagian untuk menampilkan dataset yang telah di-*training*. Pada bagian *training* data baru, pengguna bisa memilih lebih dari satu gambar yang akan di-train dan disimpan pada database gambar. Sebelum memilih proses di *training* data baru, pengguna wajib menentukan kategori gambar yang di-train. Kategori ini digunakan untuk evaluasi pada hasil pencarian gambar.



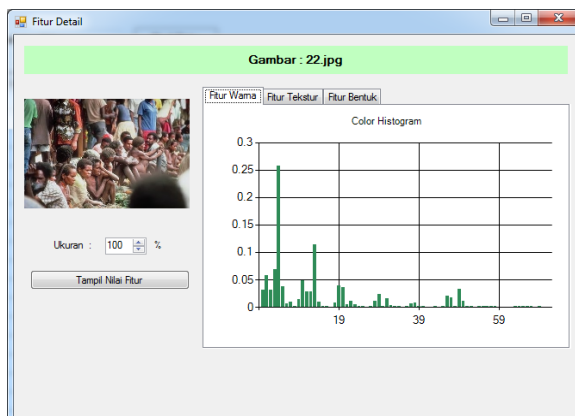
Gambar 3 Tampilan Training Aplikasi Pencarian Gambar

Pada bagian dataset *training*, pengguna dapat melihat jumlah dataset yang telah berhasil di-train. Pada Gambar 3 ditunjukkan jumlah dataset berupa 600 gambar. Pengguna dapat melihat rincian dari gambar per kategori dengan memilih tombol “tampilkan data hasil *training*”. Tampilan data *training* ditunjukkan pada Gambar 4. Pengguna dapat menampilkan data berdasarkan kategori gambar.

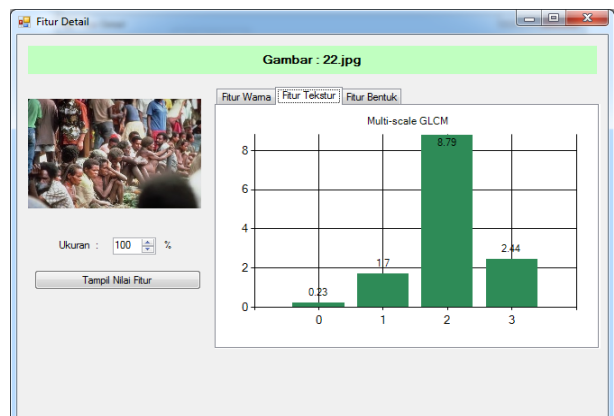


Gambar 4 Tampilan Data Gambar yang telah di Training

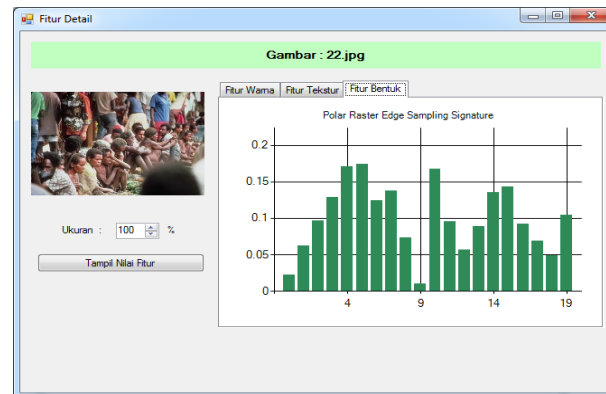
Selain itu, pengguna dapat menghapus dataset yang telah berhasil di-train dengan memilih tombol hapus yang ada disamping gambar yang telah berhasil ditrain. Pengguna juga dapat melihat rincian fitur pada gambar yang telah di-train dengan memilih tombol “show”. Gambar 5(a-c) menunjukkan rincian fitur gambar yang dipilih oleh pengguna. Pengguna dapat melihat penyebaran fitur dalam bentuk diagram batang untuk fitur warna, fitur tekstur dan fitur bentuk berdasarkan pilihan skala yang diinginkan.



(a)



(b)

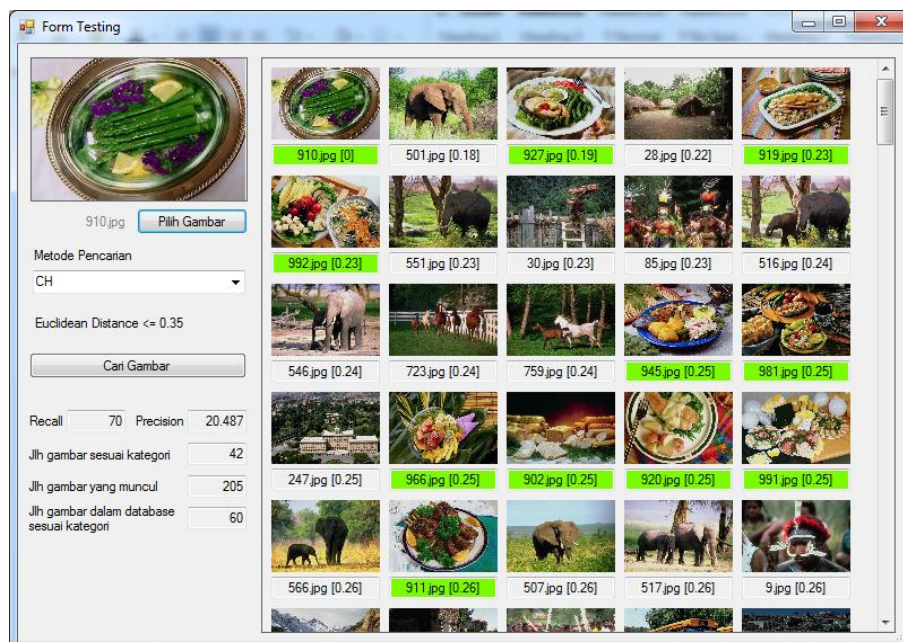


(c)

Gambar 5. Tampilan Rincian Fitur (a) Warna (b) Tekstur (c) Bentuk Gambar

3. Tampilan *testing*.

Tampilan *testing* akan muncul ketika pengguna memilih tombol *testing* pada tampilan awal aplikasi. Tampilan *testing* ditunjukkan pada Gambar 6.

**Gambar 6. Tampilan Pencarian Gambar**

Tampilan pencarian gambar terdiri dari dua bagian utama yaitu bagian untuk setting-an pencarian gambar dan evaluasi hasil pencarian, serta bagian untuk menampilkan gambar hasil pencarian gambar. Pengguna diberikan enam pilihan metode pencarian gambar yaitu metode warna (CH), tekstur (MSGLCM), bentuk (PRESS), warna dan tekstur (CHMSGLCM), warna dan bentuk (CHPRESS), tekstur dan bentuk (MSGLCMPRESS) dan warna-tekstur-bentuk (CHMSGLCMPRESS). Nilai threshold pada setiap metode telah ditentukan berdasarkan hasil pengujian yang dilakukan dengan metode *Leave-one-out-Cross-Validation* (LOOCV). Gambar hasil pencarian dengan label berwarna hijau menunjukkan bahwa gambar yang tampil merupakan gambar pada kategori yang sama dengan gambar query yang dimasukkan oleh pengguna.

3.2 Pembahasan

Pengujian dilakukan untuk dataset gambar standar Wang yang terdiri dari 10 kategori utama, yaitu Afrika, Pantai, Bangunan, Bus, Dinosaur, Gajah, Bunga, Kuda, Gunung, dan Makanan. Setiap

kategori gambar terdiri dari 60 gambar. Setiap gambar menggunakan format JPG dan memiliki ukuran 384 x 256 piksel. Hasil fitur yang dihasilkan setelah melakukan *training* terhadap 600 gambar dari dataset Wang dirincikan sebagai berikut:

1. Jumlah fitur warna untuk masing-masing skala: 43.200 data. Pada pengujian, skala gambar yang dibuat adalah gambar dengan skala 50%, 60%, 70%, 80%, 90%, 100%, 110%, 120% dan 130%. Jadi total fitur warna yang diperoleh: 388.800 data fitur.
2. Jumlah fitur tekstur untuk masing-masing skala: 2.400 data. Pada pengujian, skala gambar yang dibuat adalah gambar dengan skala 50%, 60%, 70%, 80%, 90%, 100%, 110%, 120% dan 130%. Jadi total fitur warna yang diperoleh: 21.600 data fitur.
3. Jumlah fitur bentuk untuk masing-masing skala: 12.000 data. Pada pengujian, skala gambar yang dibuat adalah gambar dengan skala 50%, 60%, 70%, 80%, 90%, 100%, 110%, 120% dan 130%. Jadi total fitur warna yang diperoleh: 108.000 data fitur.

3.2.1 Penentuan Nilai Threshold

Nilai threshold untuk masing-masing metode pencarian akan dievaluasi dan ditentukan berdasarkan hasil perhitungan presisi, recall dan F-measure yang lebih baik. Pengujian ini dilakukan dengan mengecek setiap gambar pada kategori yang sama dan skala yang sama dengan metode pencarian yang sama untuk mendapatkan nilai ambang yang lebih baik ketika dicari pada database. Proses pengujian threshold dimulai dari nilai 0 dan ditingkatkan secara linear dengan *granularity* 0.05 sampai ditemukan *local-best* dari nilai *f-measure*. Rangkuman hasil dari pengujian penentuan nilai threshold terbaik dapat dilihat dari Tabel 1.

Tabel 1 Rangkuman Pengujian Penentuan Nilai Threshold

	CH	MSGLCM	PRESS	CHMSGLCM	CHPRESS	MSGLCMPRESS	CHMSGLCMPRESS
Threshold Terbaik	0.35	0.8	0.15	0.65	0.25	0.55	0.45
F-Measure Terbaik	47.003	33.771	34.031	38.738	45.744	35.017	42.403

3.2.2 Pengaruh Skala Gambar pada Pencarian

Gambar pada dataset telah dilakukan proses *scaling* dalam pengujian. Skala yang diuji terdiri dari gambar dengan skala 50%, 60%, 70%, 80%, 90%, 100%, 110%, 120%, dan 130%. Nilai rata-rata f-measure untuk semua kategori dibandingkan untuk menentukan apakah perubahan skala memberikan efek positif dalam meningkatkan keakuratan pencarian gambar. Untuk mendapatkan hasil f-measure yang maksimal, perbandingan bukan hanya dilakukan pada satu nilai threshold, namun pada semua nilai threshold yang diuji pada Pengujian Nilai Threshold. Hasil dari pengujian pengaruh skala gambar pada proses pencarian terbaik di aplikasi CBIR dapat dirangkum pada Tabel 2.

Tabel 2 Rangkuman Pengujian Pengaruh Skala Gambar

	CH	MSGLCM	PRESS	CHMSGLCM	CHPRESS	MSGLCMPRESS	CHMSGLCMPRESS
Skala Terbaik	100%	100%	50%	100%	100%	100%	100%
F-Measure Terbaik	47.339	36.141	35.245	42.059	46.249	38.418	43.629

4. KESIMPULAN

Berdasarkan hasil pengujian pencarian gambar pada dataset yang diamati, maka dapat ditarik kesimpulan sebagai berikut:

1. Pencarian Gambar dengan nilai F-measure terbaik diperoleh dari fitur warna (47.339), kombinasi warna dan bentuk (46.249), dan kombinasi warna, tekstur dan bentuk (43.629). Hasil ini menunjukkan fitur warna, tekstur dan bentuk dapat digunakan dalam proses pencarian gambar.
2. Perubahan skala pada gambar belum menunjukkan peningkatan keakuratan hasil pencarian dengan kombinasi fitur warna dan tekstur. Hal ini ditunjukkan dengan tidak adanya peningkatan nilai F-measure pada metode fitur warna dan tekstur ketika skala gambar diubah lebih kecil dan besar.

3. Perubahan skala pada gambar menunjukkan adanya peningkatan keakuratan hasil pencarian dengan fitur bentuk. Hal ini ditunjukkan oleh peningkatan nilai F-measure pada pengujian dengan fitur PRESS, dimana nilai F-measure menjadi 35.245 (skala gambar 50%) dibandingkan dengan F-measure = 34.09 pada saat gambar tidak berubah (skala gambar 100%)

5. SARAN

Beberapa saran yang dapat diusulkan Penulis ketika melakukan penelitian ini:

1. Perlunya penelitian lanjutan yang menentukan nilai threshold berdasarkan metode iterasi untuk mendapatkan nilai threshold yang lebih baik.
2. Perlu kajian lebih jauh mengenai pengaruh dari fitur tekstur dalam pencarian gambar karena dari hasil pengujian, fitur tekstur kurang memberikan pengaruh yang signifikan dalam meningkatkan kualitas hasil pencarian gambar yang akurat.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada STMIK Mikroskil yang telah memberikan dukungan dana melalui Hibah Kompetisi Penelitian Internal STMIK Mikroskil No. 403/ST.45/01/PN/2015, sehingga penelitian dapat selesai dengan baik.

DAFTAR PUSTAKA

- [1] Smeulders, A. W. M., Worring, M., Santini, S., Gupta, A., Jain, R., 2000, Content-Based Image Retrieval at the End of the Early Years, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 22, No.12, pp. 1349-1379
- [2] Singh, B. dan Ahmad, W., 2014, Content Based Image Retrieval: A Review Paper, International Journal of Computer Science and Mobile Computing, Vol 3, Issue 5.
- [3] Ali, Nursabillilah Mohd, Rashid, Nahrul Khair Alang Md, and Mustafah, Yasir Mohd, 2013, Performance Comparison between RGB and HSV Color Segmentations for Road Signs Detection Applied Mechanics and Materials, Vol. 393, pp 550-555
- [4] Arthi, K., Vijayaraghavan, J., 2013, Content Based Image Retrieval Algorithm using Colour Models, International Journal of Advanced Research in Computer and Commucation Engineering, Vol. 2, Issue 3
- [5] Halim, A., Hardy, Dewi, C., Angkasa, S., 2013, Aplikasi Image Retrieval menggunakan Kombinasi Metode Color Moment dan Gabor Texture, Jurnal SIFO Mikroskil, vol 14 no 2, pp. 109-117
- [6] Halim, A., Hardy, Yufandi, A., Fiana, 2014, Aplikasi Content Based Image Retrieval dengan Fitur Warna dan Bentuk, Jurnal SIFO Mikroskil, vol 15 no 2, pp. 83-90
- [7] John, Deepak, Tharani, S.T., Sreekumar K, 2014, Content Based Image Retrieval using HSV-Color Histogram and GLCM, International Journal of Advance Research in Computer Science and Management Studies, Volume 2, Issue 1, pp. 246-253
- [8] Siqueira, F. R. D., Schwartz, W. R., Pedrini, H., 2012, Multi-Scale Gray Level Co-Occurrence Matrices for Texture Description, Institute of Computing, University of Campinas.
- [9] Halim, A., Gohzali, H., Sin, I., Wijaya, K. 2015, 1. Content Based Image Retrieval dengan Algoritma Sobel's Edge Detection, Seminar Nasional Informatika, Agustus 22.
- [10] Mathew, S. P., Balas, V. E., Zachariah K. P., Samuel, P., 2014, A Content-based Image Retrieval System Based on Polar Raster Edge Sampling Signature, Acta Polytechnica Hungarica, volume 11, nomor 3.
- [11] Deepak, J., Sreekumar, K., Tharani, S. T., 2014, Content Based Image Retrieval using HSV-Color Histogram and GLCM, International Journal of Advance Research in Computer Science and Management Studies, volume 2, issue 1.