

Pencarian Citra Digital Berbasis Konten dengan Ekstraksi Fitur HSV, ACD, dan GLCM

Muhammad Dendy Agaputra^{#1}, Ken Ratri Retno Wardani^{#2}, Elisafina Siswanto^{#3}

Departemen Teknik Informatika - Institut Teknologi Harapan Bangsa

Jalan Dipatiukur no. 83 – 84 Bandung

¹dendy_agaputra@gmail.com

²ken_ratri@ithb.ac.id

³elisafina@ithb.ac.id

Abstrak—Pencarian citra berbasis konten adalah teknik untuk mencari citra yang piksel-pikselnya memiliki karakteristik sama atau mendekati dengan sebuah sekumpulan citra. Penelitian ini menggunakan warna dan tekstur sebagai ekstraksi fitur dari sebuah citra. Untuk warna digunakan dua bentuk pendekatan, yaitu HSV (*Hue, Saturation, Value*) dan ACD (*Average Color Dominance*). Sedangkan tekstur menggunakan metode *Gray Level Co-occurrence Matrices*. Kemudian untuk proses pencocokan antara citra *query* dengan citra target yang ada pada sekumpulan citra dilakukan perhitungan jarak (*Euclidean Distance*) dari citra *query* dengan citra target pada basis data citra. Dari hasil pengujian didapatkan bahwa pencarian citra melalui ekstraksi warna (HSV, ACD) dan tekstur (GLCM) menghasilkan hasil pencarian yang lebih akurat dibanding dengan hanya ekstraksi warna saja atau ekstraksi tekstur saja. Selain itu, citra yang mengandung *Point of Interest* (POI) tunggal jelas lebih baik dalam pencarian dibandingkan dengan citra yang memiliki banyak POI.

Kata Kunci — Pencarian citra berbasis konten, warna, tekstur, HSV, ACD, GLCM

Abstract—*Content-based image retrieval is a technique to search for a set of images based on its pixels that have the same characteristics with query image. This research used color and texture as feature extraction of an image. For color, we use two approaches, namely the HSV (Hue, Saturation, Value) and ACD (Average Color Dominance), while for the texture, we use the Gray Level Co-occurrence Matrices method. The matching process between the query image with the target image is done by calculating the distance by using Euclidean Distance. The experimental results showed that the extraction of image retrieval through color (HSV, ACD) and texture (GLCM) produces more accurate results compared with extraction using color or texture only. In addition, images that have a single and clear Point of Interest (POI) have better result than the image that has a lot of POI.*

Keywords — *Content-based image retrieval, color, texture, HSV, ACD, GLCM*

I. PENDAHULUAN

Pencarian citra berbasis konten adalah aplikasi yang menggunakan pengolahan citra untuk mencari citra tertentu yang mirip terhadap basis data yang besar. Istilah

“berbasis konten” berarti pencarian tersebut akan menganalisis konten yang sebenarnya yang ada di dalam sebuah citra bukan melalui metadata seperti kata kunci, *tags* dan/atau deskripsi yang terkait dengan citra. Kata “konten” di sini berarti warna, bentuk, tekstur atau semua informasi yang terkandung di dalam data citra itu sendiri.

Konten yang terkandung dalam citra digital adalah warna, bentuk dan tekstur. Pencarian citra berbasis konten diperlukan karena pada umumnya mesin pencari citra berbasis web masih mengandalkan metadata untuk melakukan pencarian citra dan hal tersebut dapat menyebabkan terjadi banyak kesalahan hasil pencarian. Manusia yang secara manual memasukkan kata kunci untuk citra di dalam basis data yang besar adalah hal yang tidak efisien, mahal dan bisa jadi tidak mewakili semua kata kunci yang mendeskripsikan citra. Oleh karena itu, sebuah sistem yang dapat menyaring citra berdasarkan konten akan memberikan pengindeksan yang lebih baik dan hasil yang lebih akurat.

Warna adalah fitur visual yang paling sering digunakan dalam pencarian citra berbasis konten [3]. Sebuah citra digital adalah sebuah set dari piksel – piksel yang mana tiap piksel merepresentasikan warna. Model warna (*color model*) adalah sebuah cara untuk merepresentasikan warna yang diindera manusia dalam komputasi.

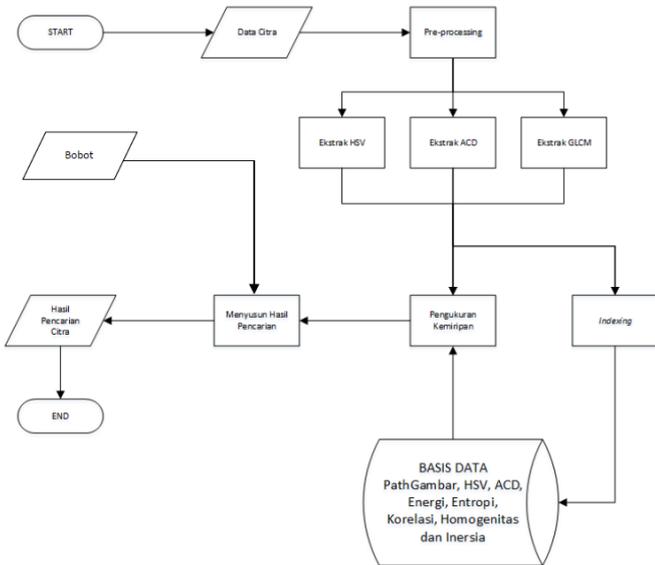
Menghitung besaran tekstur dapat dilakukan melalui metode - metode seperti *co-occurrence matrices*, *gabor filters* atau *tamura*. Dari hasil penelitian “*Evaluation of Texture Features for Content-Based Image Retrieval*” yang dilakukan oleh Peter Howarth dan Stefan Ruger, terlihat bahwa metode *co-occurrence matrices* memiliki performa terbaik dari sisi akurasi retrieval. Oleh karena itu, penelitian ini memakai metode tersebut untuk melakukan ekstraksi fitur.

II. RANCANGAN SISTEM

Alur kerja sistem diperlihatkan oleh Gambar 1.

A. Data Sampling

Data citra yang digunakan dalam penelitian ini berasal dari <http://wang.ist.psu.edu/docs/related/>. Data *sampling* yang disediakan sebanyak 1000 citra digital, terdiri dari 10 kategori dan masing-masing kategori memiliki 100 data citra digital.



Gambar 1. Flowchart Pencarian Citra

Kategori ini terbagi dalam *Africa people and villages*, pantai, bangunan, bus, dinosaurus, gajah, bunga, kuda, pegunungan dan glesler, makanan. Data citra tersebut memiliki resolusi 384x256 piksel, berformat .jpg dan memiliki 24 bit kedalaman warna.

B. Preprocessing

Sebelum dilakukan proses *matching* diperlukan adanya tahap *preprocessing* yang mana pada penelitian kali ini data citra *query* diubah ukurannya menjadi 160x240 piksel (panjang x lebar) untuk mempercepat proses *matching* namun tidak mengurangi kualitas hasil pencarian. Setelah diubah ukurannya, data citra tersebut dibagi menjadi 6 blok atau 2 baris 3 kolom dengan ukuran yang sama.

Nilai warna dan tekstur akan dikalkulasi untuk tiap blok secara terpisah dengan blok lainnya dan disimpan secara terpisah untuk masing - masing blok. Memartisi data citra dapat dilihat seperti Gambar 2.

Setelah dilakukan *preprocessing* pada data citra, sistem kemudian mulai mengekstrak fitur warna (HSV, ACD) dan tekstur (energi, entropi, korelasi, homogenitas, dan kontras) secara bersamaan untuk tiap – tiap blok data citra. Ketujuh nilai hasil ekstraksi fitur warna dan tekstur tersebut dibandingkan dengan ketujuh nilai ekstraksi fitur warna dan tekstur data citra yang telah disimpan di basis data. Cara membandingkan nilai tersebut adalah dengan menggunakan perhitungan *Euclidean Distance*.

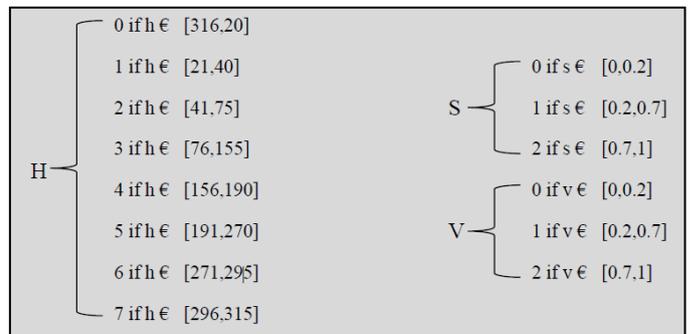
Untuk menggabungkan ketujuh *list* ekstraksi menjadi satu *list* yang terurut berdasarkan kemiripan citra, jumlah ketujuh jarak tersebut digunakan untuk merepresentasikan kemiripan. Jarak akan bernilai relatif lebih kecil jika kedua citra memiliki kemiripan. Sebaliknya, jarak akan bernilai relatif lebih besar jika kedua citra tidak memiliki kemiripan dari sisi warna dan tekstur.

C. Ekstraksi Fitur HSV



Gambar 2. Citra 2x3 Blok

HSV merupakan salah satu cara untuk mendefinisikan warna yang didasarkan pada roda warna. Hue mengukur sudut sekitar roda warna (merah pada 0 derajat, 120 derajat di hijau, biru, di 240 derajat). *Saturation* yang menunjukkan pada radius roda warna sehingga menunjukkan proporsi antara gelap (pusat) untuk warna ke putih murni (di luar). *Value* menunjukkan nilai kecerahan. *Hue* memiliki nilai antara 0 hingga 360 (derajat), *saturation* and *value* berkisar dari 0 hingga 100%.



Nilai HSV yang dijadikan tolak ukur dalam penelitian ini dilakukan tahap normalisasi kedalam angka yang lebih sederhana. Hal ini bertujuan untuk mempersingkat waktu komputasi, namun tidak mengurangi akurasi pencarian secara signifikan. Kuantisasi dilakukan dengan persamaan di atas.

Vektor 3 dimensi HSV tersebut dikonversi menjadi vektor 1 dimensi dengan tetap mempertimbangkan bobot dari masing – masing nilai komponen HSV [1]. Tujuannya agar nilai HSV ini dapat dicari jaraknya melalui *Euclidean Distance*. Persamaan yang digunakan adalah [1]:

$$G = (9*H) + (3*S) + V \quad (1)$$

D. Ekstraksi Fitur ACD

ACD pada dasarnya adalah penggunaan lebih lanjut dari model warna RGB. ACD adalah teknik untuk menggambarkan warna dominan yang ada di suatu image maupun subimage. Untuk mendapatkan nilai ACD, pertama-tama adalah dengan menguantisasi data citra menjadi warna yang lebih sedikit (dalam penelitian ini menggunakan 11 warna). Dalam penelitian ini, metode *Median Cut Quantizer* yang digunakan untuk menyederhanakan warna.

Lalu menghitung banyaknya kemunculan masing-masing warna pada tiap piksel. Penghitungan ini menggunakan prinsip *4-neighbour* yang artinya piksel warna hanya dihitung jika memiliki warna yang sama dengan keempat piksel tetangganya (atas, kanan, bawah, kiri). Setelah mendapatkan banyaknya kemunculan tiap warna, sistem mengambil 3 warna yang paling sering muncul dan diambil nilai rata-rat RGB.

Dari perhitungan tersebut, dapat ditarik kesimpulan warna apa saja yang mendominasi citra. Untuk menunjang akurasi, peneliti tidak mengambil satu nilai warna yang paling banyak muncul, melainkan mengambil tiga warna dominan lalu diambil nilai rata-rata Hasil dari rata – rata tersebut yang kemudian disimpan kedalam basis data sebagai salah satu fitur warna bersama dengan fitur warna lainnya, yaitu HSV.

```

begin
    • Resize citra keukuran 160x240
    • Partisi citra kedalam n blok

    forall n blok citra
        • Kuantisasi warna menjadi 8 warna
        • Hitung kemunculan warna menggunakan konsep 4-neighbour
        • Kuantisasi vector 3 dimensi RGB menjadi 1 dimensi dengan cara merata-ratakan RGB ((R+G+B)/3)
        • Kalkulasi rata-rata RGB 3 warna yang paling banyak muncul
        • Simpan nilai rata-rata RGB (ACD value) kedalam basis data untuk blok citra ini
    endfor
end
    
```

E. Ekstraksi Fitur GLCM

Pada analisis tekstur secara statistis, fitur tekstur dihitung berdasarkan distribusi statistik dari kombinasi intensitas piksel pada posisi tertentu relatif terhadap lainnya dalam suatu matriks citra. Bergantung pada jumlah piksel atau titik intensitas dalam masing-masing kombinasi, dibedakan adanya statistik orde-pertama, statistik orde-kedua dan statistik orde-lebih-tinggi (*higher-order statistics*).

Metode GLCM (*gray-level co-occurrence matrix*) adalah salah satu cara mengekstrak fitur tekstur statistik orde-kedua [2]. GLCM, yang disebut juga *Grey Tone Spatial Dependency Matrix*, adalah tabulasi mengenai frekuensi atau seberapa seringnya kombinasi nilai kecerahan piksel yang berbeda posisinya terjadi dalam suatu citra [4]. GLCM dihitung dengan algoritma sebagai berikut:

```

begin
forall gray levels a,b ∈ image
    P(a,b)=0;
endfor
forall location (x,y)
    ∈ image and (x+Δx,y+Δy)
    ∈ image
    a=image(x,y);
    b=image(x+Δx,y+Δy);
endfor
end
    
```

Berikut ini adalah gambaran pembentukan GLCM atas citra dengan 4 tingkat keabuan (*gray level*) pada jarak $d = 1$ dan arah 0° . Gambar 3 menunjukkan arah dan jarak yang dapat dipakai dalam menghitung GLCM sebuah citra. Dari piksel di tengah (x) piksel 1 menunjukkan arah = 0° dengan jarak $d = 1$; piksel 2 arah = 45° dengan jarak $d = 1$; piksel 3 arah = 90° dengan jarak $d = 1$; dan piksel 4 arah = 135° dengan jarak $d = 1$, seperti yang pada Gambar 4. Matriks *co-occurrence* menangkap sifat tekstur tetapi tidak secara langsung dapat digunakan sebagai alat analisis, misalnya membandingkan dua tekstur.

Data ini harus disarikan lagi agar didapatkan angka-angka yang bisa digunakan untuk mengklasifikasi tekstur. Ciri atau fitur statistik GLCM antara lain:

1) *Kontras*

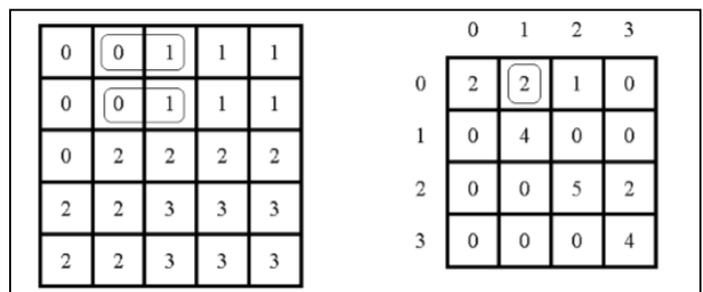
Perhitungan kontras berkaitan dengan jumlah keberagaman intensitas keabuan dalam citra.

$$\text{kontras} = \sum_{a,b=0}^{n-1} |a - b|^2 P_{a,b} \tag{2}$$

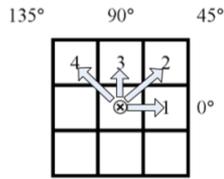
2) *Homogenitas*

Secara matematis, homogenitas GLCM adalah invers dari kontras GLCM, yaitu keseragaman intensitas keabuan pada citra.

$$\text{Homogenitas} = \sum_{a,b=0}^{n-1} \frac{P_{a,b}}{1 + (a - b)^2} \tag{3}$$



Gambar 3. (kiri) Contoh Citra dengan 4 Tingkat Keabuan. (kanan) GLCM pada Jarak 1 Arah 0°



Gambar 4. Arah dan Jarak GLCM

3) Energi

Energi menyatakan ukuran konsentrasi pasangan dengan intensitas keabuan tertentu pada matriks.

$$\text{Energi} = \sum_{a,b=0}^{n-1} P_{a,b}^2 \quad (4)$$

4) Entropi

Entropi digunakan untuk mengukur keteracakan dari distribusi intensitas.

$$\text{Entropi} = \sum_{a,b=0}^{n-1} P_{a,b} \log_2 (P_{a,b}) \quad (5)$$

5) Korelasi

Menyatakan ukuran hubungan ketergantungan piksel terhadap piksel tetangga dalam citra.

$$\text{Korelasi} = \sum_{a,b=0}^{n-1} P_{a,b} \left[\frac{(a - \mu_x)(b - \mu_y)}{\sigma_x \sigma_y} \right]$$

$$\mu_x = \sum_{a=0}^{n-1} a \sum_{b=0}^{n-1} P_{a,b}$$

$$\mu_y = \sum_{b=0}^{n-1} b \sum_{a=0}^{n-1} P_{a,b} \quad (6)$$

$$\sigma_x = \sum_{a=0}^{n-1} (a - \mu_x)^2 \sum_{b=0}^{n-1} P_{a,b}$$

$$\sigma_y = \sum_{b=0}^{n-1} (b - \mu_y)^2 \sum_{a=0}^{n-1} P_{a,b}$$

Seperti yang tertera pada persamaan (6), perlu mencari nilai mean (μ) dan standar deviasi (σ) sebelum mencari nilai korelasi. *Mean* adalah rata-rata dari suatu sebaran nilai intensitas citra keabuan. Sedangkan standar deviasi adalah menunjukkan sebaran nilai piksel pada bidang citra. Proses ekstraksi GLCM dapat dilihat pada Gambar 5.



Gambar 5. Flowchart Ekstraksi GLCM

F. Pengukuran Kemiripan

Kemiripan pada penelitian ini direpresentasikan oleh nilai fitur warna dan tekstur yang sudah disimpan di basis data sebelumnya. Untuk dapat melakukan perbandingan data yang telah diekstraksi dari citra, dilakukan perhitungan masing – masing fitur untuk tiap blok, tiap citra. Sebagai contoh, dilakukan komputasi jarak fitur HSV menggunakan *Euclidean Distance* pada blok#1 pada citra *query* dengan blok#1 pada citra pertama dibasis data, lalu dilakukan komputasi pada blok#2 citra *query* dengan blok#2 citra pertama pada basis data dan selanjutnya hingga blok terakhir, seperti pada Gambar 6.

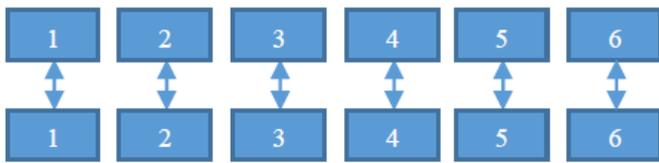
Berikut ini adalah contoh penggunaan *Euclidean Distance* pada fitur HSV, ACD, energi, entropi, korelasi, homogenitas, dan kontras. Gambar 7 memperlihatkan contoh perhitungan jarak kemiripan antara suatu blok citra *query* dengan blok citra sumber. Untuk perhitungan jarak kemiripan pada fitur GLCM dilakukan perhitungan serupa. Setelah didapat masing – masing jarak pada semua fitur per blok per citra, dilakukan pengurutan ranking untuk menentukan citra mana yang paling mirip dan citra mana yang lebih tidak mirip.

Nilai jarak masing-masing fitur pada citra *query* dengan citra target disusun secara *ascending* untuk menunjukkan citra-citra mana saja yang jaraknya kecil kepada citra *query* dan yang besar.

Pertama-tama, list1, list 2, list3, list4, list5, list6 dan list7 adalah sebuah List atau Array yang menampung jarak antara citra *query* dengan citra target (list1=hsv, list2=acd, list3=entropi dst). Sebagai contoh, blok #1 citra#32 terletak di posisi 15 pada list1, posisi 55 pada list2, posisi 60 pada list3, posisi 19 pada list4, posisi 150 pada list5, posisi 345 pada list6 dan posisi 22 pada list7. Penjumlahan seluruh peringkat pada blok#1 citra#32 menghasilkan $(15*a)+(55*b)+(60*c)+(19*d)+(150*e)+(345*f)+(22*g) = 666$ (a,b,c,d,e,f,g adalah bobot masing-masing fitur yang diisi oleh pengguna). Nilai ini dikalkulasikan pada tiap blok tiap citra untuk mendapatkan seberapa dekat blok#1 pada citra *query* dengan blok#1 citra sumber (basis data).

III. PENGUJIAN

Pengujian akurasi dilakukan dengan mengambil data uji dari tiap kategori lalu mengkalkulasi tingkat relevansi hasilnya dengan banyaknya citra kategori tersebut dibasis data. Sebagai contoh, citra 774.jpg dijadikan data uji yang menghasilkan data citra pencarian. Dari hasil pencarian tersebut, dihitung banyaknya citra yang berasal dari kategori



Gambar 6. Komparasi antar blok citra

(13)	(6)	(53)	(16)	(32)
(28)	(45)	(59)	(12)	(24)
(46)	(46)	(9)	(11)	(39)
(4)	(54)	(6)	(15)	(18)
(31)	(31)	(29)	(15)	(34)

(8)	(51)	(16)	(37)	(42)
(28)	(55)	(59)	(12)	(24)
(4)	(41)	(9)	(10)	(34)
(25)	(63)	(31)	(19)	(11)
(2)	(27)	(19)	(15)	(33)

$$d(i, j) = \sqrt{|13 - 8|^2 + |6 - 51|^2 + |53 - 16|^2 + \dots + |34 - 33|^2}$$

= 89.69392398596462

Gambar 7. a) Vektor Nilai Fitur Citra *query*. b) Vektor Nilai Fitur Citra Basis Data. c) Perhitungan Jarak antara Kedua Vektor

yang sama (dalam contoh ini citra 774.jpg berkategori kuda) yaitu 71 data citra. Angka tersebut dibagi dengan banyaknya data citra di kategori kuda (100) kemudian dikali 100% agar mendapatkan persentase keakuratan pencarian.

Skenario pengujian dibagi menjadi tiga bagian, yang pertama menguji pencarian ditiap kategori berdasarkan ekstraksi fitur warna, dilanjutkan dengan pengujian melalui ekstraksi fitur tekstur dan yang terakhir pengujian pencarian citra melalui ekstraksi fitur tekstur dan warna.

Untuk masing – masing skenario, penelitian ini melakukan *query* ke-1000 data citra yang menjadi data sampling terhadap 1000 data citra yang telah diindeks. Hal ini bisa dilakukan dengan membuat potongan program yang melakukan iterasi terhadap semua data sampling kemudian menguji akurasi pencarian citranya. Data uji yang ditampilkan pada penelitian ini merupakan data yang memiliki akurasi paling tinggi dikategorinya masing-masing.

Akurasi citra didapatkan dengan membandingkan nama *file* citra *query* dengan nama *file* citra yang dicari. Perlu diketahui, bahwa nama *file* citra pada data sampling di penelitian ini mewakili kategorinya. Sebagai contoh, citra dengan nama 0.jpg–99.jpg merupakan kategori *Africa people and villages*, 100.jpg–199.jpg merupakan kategori pantai, 200.jpg–299.jpg merupakan kategori bangunan dan seterusnya.

Kategori dari citra *query* kemudian dibandingkan dengan seluruh kategori dari citra hasil pencarian. Nilai akurasi merupakan jumlah kategori citra yang cocok dibagi dengan total citra hasil pencarian. Hasil akurasi pengujian untuk setiap kategori dapat dilihat pada Tabel I.

IV. KESIMPULAN DAN SARAN

Berikut adalah kesimpulan dari penelitian yang dilakukan:

TABEL I
HASIL PENGUJIAN

Kategori	Akurasi (%)		
	Tekstur	Warna	Warna & Tekstur
<i>Africa people and villages</i>	32	33	40
Pantai	42	42	48
Bangunan	31	22	32
Bus	51	5	41
Dinosaur	77	33	89
Gajah	34	41	39
Bunga	64	45	80
Kuda	65	65	76
Pegunungan dan glistier	30	48	38
Makanan	32	36	36

- Aplikasi dapat mengekstrak fitur HSV, ACD dan GLCM citra *query* lalu menampilkan data citra yang mempunyai jarak terdekat terhadap fitur tersebut menggunakan *Euclidean Distance*.
 - Secara keseluruhan, pencarian citra melalui ekstraksi warna (HSV, ACD) dan tekstur (GLCM) menghasilkan hasil pencarian yang lebih akurat dibanding dengan hanya ekstraksi warna saja atau ekstraksi tekstur saja.
 - Pengguna dapat mengatur besarnya bobot masing-masing fitur (warna dan tekstur) sehingga hasil pencarian citra sesuai dengan keinginan pengguna tersebut.
 - Citra yang mengandung *Point of Interest* (POI) tunggal dan jelas lebih baik dalam pencarian dibanding citra yang memiliki banyak POI. POI adalah obyek yang menjadi pusat dari sebuah citra.
 - Beberapa optimasi yang dilakukan saat implementasi kedalam perangkat lunak, seperti mengubah sistem penyimpanan menggunakan *file* dan *multi-threading* pada beberapa proses menghasilkan waktu eksekusi program yang lebih cepat dan kebutuhan ruang penyimpanan yang efisien.
- Berikut adalah saran yang perlu dipertimbangkan untuk pengembangan penelitian kedepan :
- Penggunaan *multi-thread* pada saat memuat seluruh basis data mengalami *bottle-neck* pada saat membaca *hard disk*. Hal tersebut dikarenakan peneliti menggunakan *sequential drive* yang hanya dapat membaca satu berkas dalam satu waktu. Akan lebih baik jika kedepannya perangkat lunak ini ditanam disuatu sistem yang menggunakan *random access drive*, seperti SSD (*Solid State Drive*).
 - Diharapkan aplikasi ini dapat meningkatkan akurasi pencarian citra digital berbasis konten dengan menambah metode / fitur lainnya yang dapat diekstrak dari sebuah citra digital.
 - Berkas .txt yang dibuat tidak aman dari perubahan eksternal, dengan kata lain siapapun dapat

memanipulasi data ini. Sebaiknya digunakan teknik tertentu pada data tersebut agar tidak dapat dimanipulasi pengguna selain melalui sistem ini.

REFERENSI

- [1] Ch, Dr B Kavitha, Prabhakara Rao, and Dr A. Govardha, "Image retrieval based on color and texture features of the image sub-blocks.," *International Journal of Computer Applications*, vol. 15, no. 7, 2011.
- [2] Fritz Albrechtsen, "Statistical Texture Measures Computed from Gray Level Cooccurrence Matrices.," Image Processing Laboratory, Department of Informatics, University of Oslo, 1995.
- [3] Henning Müller, Nicolas Michoux, Davidand Bandon, and Antoine Geissbuhler, "A review of content-based image retrieval systems in medical applications—clinical benefits and future directions.," *International Journal of Medical Informatics*, vol. 73, no. 1, pp. 1-23, 2004.
- [4] Mryka Hall-Beyer. (2007) The GLCM Tutorial Home Page. [Online]. <http://www.fp.ucalgary.ca/mhallbey/tutorial.htm>.

Dendy Argaputra, lahir di Dumai pada tahun 1991, menerima gelar Sarjana Teknik dari Institut Teknologi Harapan Bangsa pada tahun 2013 jurusan Teknik Informatika.

Ken Ratri Retno Wardani, lahir di Bandung pada tahun 1968, menerima gelar Sarjana Teknologi Informasi dari Institut Teknologi Bandung pada tahun 2004 dari Departemen Teknik Elektro. Minat penelitian adalah pada bidang Pengolahan Citra.

Elisafina Siswanto, lahir di Bandung pada tahun 1989, menerima gelar Sarjana Teknik dari Institut Teknologi Harapan Bangsa pada tahun 2011 jurusan Teknik Informatika. Saat ini sedang menempuh studi Magister di Institut Teknologi Bandung jurusan Informatika dan aktif sebagai pengajar di Departemen Teknik Informatika, Institut Teknologi Harapan Bangsa di Bandung. Minat penelitian adalah pada bidang Pembelajaran Mesin dan Pemrosesan Bahasa Alami.