**Aplikasi Temu Kembali Citra Dengan Fitur Warna dan GLCM Serta Klasterisasi *K-Means* Berbasis Java Pada Fakultas Teknologi Informasi Universitas Budi Luhur**

**Hariya Saputra1), Achmad Solichin2)**

*Program Studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Budi Luhur*

*Jl. Raya Ciledug, Petukangan Utara, Kebayoran Lama, Jakarta Selatan 12260*

*Telp. (021) 5853753, Fax. (021) 5866369*

*E-mail :* [*hariya.saputra03@gmail.com*](mailto:hariya.saputra03@gmail.com)

**ABSTRAK**

*Teknik pencarian informasi biasanya dilakukan dengan menggunakan metode berbasiskan teks (text-based). Metode tersebut juga dapat diterapkan pada data berupa citra. Permasalahan yang kerap terjadi adalah kata kunci yang terdapat pada citra tidak relevan dengan visual citra. Tentu saja hal ini berpengaruh pada keakuratan hasil pencarian. Sistem temu kembali citra berdasarkan isi atau Content-Based Image Retrieval (CBIR) merupakan sistem yang bertujuan untuk mencari sejumlah citra berdasarkan fitur yang ada. Dalam tugas akhir ini dibangun sistem temu kembali citra yang didasarkan pada fitur warna dan tekstur. Ekstraksi fitur warna dilakukan dalam bentuk histogram, yaitu dengan menghitung nilai frekuensi warna tiap selang warna tertentu. Sedangkan untuk ektraksi fitur tekstur menggunakan metode Grey-Level Co-occurence Matrix (GLCM). Nilai fitur tekstur yang digunakan antara lain Energy, Entropy, Homogenity, Correlation, dan Contrast. Setiap citra yang telah memiliki nilai fitur selanjutnya diklasterisasi dengan menggunakan algoritma K-Means. Klasterisasi merupakan metode untuk mengelompokkan citra ke dalam bentuk satu atau lebih cluster berdasarkan fitur dari masing-masing citra. Tujuan dari klasterisasi dalam penelitian ini adalah untuk mempercepat proses komputasi serta meningkatkan keakuratan hasil pencarian citra. Hasil uji coba dengan menggunakan 1000 data citra (terdiri dari 10 kelompok) yang diambil dari* [*http://wang.ist.psu.edu/docs/related/*](http://wang.ist.psu.edu/docs/related/) *menunjukkan bahwa, metode yang digunakan dalam ekstraksi fitur dapat menemukan kembali citra yang relevan pada citra acuan dengan rata-rata presisi yaitu 81,4%. Sementara itu, hasil perolehan tingkat presisi dengan menggunakan clustering dapat lebih baik atau lebih buruk. Hal ini dikarenakan algoritma yang digunakan bergantung pada pembangkitan centroid awal yang dilakukan secara random.*

**Kata** **kunci** : CBIR, *Color Histogram*, GLCM, *K-Means Clustering*

1. **PENDAHULUAN**

Pada era teknologi digital saat ini, perkembangan kapasitas media penyimpanan merupakan suatu kebutuhan. Hal ini mengakibatkan jumlah *file* digital meningkat sangat pesat karena kapasitas media penyimpanan yang semakin besar. Meningkatnya jumlah *file* digital juga dapat berpengaruh terhadap proses pencarian, terutama file citra (*image*). Citra dapat mengandung banyak informasi, karena citra dapat menjelaskan beberapa kejadian. Selain itu citra juga sering digunakan untuk mempresentasikan suatu objek. Namun, pencarian citra yang biasa dilakukan yaitu berbasiskan teks memiliki kekurangan. Hal ini dikarenakan keterangan atau kata kunci yang diberikan tidak relevan dengan visual citra itu sendiri. Sehingga pencarian citra berbasiskan teks menyebabkan hasil pencarian citra tidak sesuai dengan citra yang diharapkan. Selain itu pencarian juga akan menjadi sulit apabila seseorang tidak mengetahui kata kunci dari objek tersebut.

Permasalahan ini juga dialami oleh unit Fakultas Teknologi Informasi (FTI) Universitas Budi Luhur. Unit ini memiliki aset gambar yang cukup besar. Misalnya adalah foto mengenai fasilitas kampus atau foto kegiatan seperti seminar, workshop, bazar, dan lain sebagainya. Biasanya pengambilan foto tidak dilakukan satu kali, bisa dilakukan dua kali bahkan lebih untuk setiap sesinya. Hal ini dilakukan untuk mendapatkan foto dengan kualitas terbaik. Karena itulah, jumlah *file* citra dapat meningkat sangat pesat. Hal tersebut menyebabkan pencarian dengan cara manual akan merepotkan. Ditambah lagi, foto-foto tersebut tidak diorganisir dengan baik. Selain itu, nama *file* yang tidak sesuai dengan visual foto menyebabkan pencarian berbasiskan teks tidak dapat diandalkan.

Melihat permasalahan tersebut maka dibutuhkan sebuah sistem yang mampu mengelola citra tersebut dan mencari citra secara otomatis dan akurat. Solusinya adalah dengan membuat aplikasi temu kembali citra berdasarkan isi atau *Content-Based Image Retrieval* (CBIR). CBIR merupakan sistem yang bertujuan untuk mencari sejumlah citra berdasarkan fitur yang ada. Warna dan tekstur merupakan dua fitur visual yang penting dari suatu citra [1] sehingga warna dan tekstur banyak digunakan sebagai fitur dalam aplikasi CBIR.

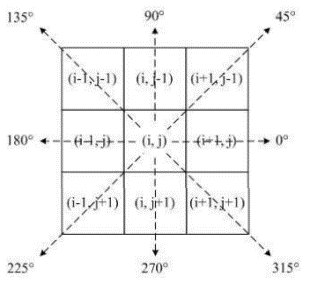
Dalam tugas akhir ini akan dibangun aplikasi temu kembali citra yang didasarkan pada fitur warna dan tekstur. Ekstraksi fitur warna dilakukan dalam bentuk histogram dan ektraksi fitur tekstur dilakukan dengan menggunakan metode *Gray-Level Co-occurence Matrix* (GLCM). Setelah ekstraksi fitur, tahap selanjutnya adalah mengelompokkan atau mengklasterisasi citra *database* dengan menggunakan algoritma *K-Means*. Dengan adanya klasterisasi, diharapkan dapat mempercepat waktu komputasi serta meningkatkan keakuratan hasil pencarian.

1. **LANDASAN TEORI**
2. ***Bicubic Inetrpolation***

*Bicubic interpolation* merupakan salah satu metode interpolasi (atau biasa disebut dengan istilah *resampling*) yang digunakan untuk meningkatkan atau mengurangi jumlah piksel dalam citra digital [2]. Interpolasi bekerja dengan menggunakan informasi yang terkandung dalam data untuk memperkirakan nilai-nilai pada titik-titik yang tidak diketahui. *Bicubic interpolation* adalah interpolasi yang canggih dan menghasilkan tepi-tepi yang halus dibandingkan dengan metode interpolasi lainnya. Metode ini menggunakan 4 x 4 piksel tetangga untuk mengambil informasi yang terkandung dalam data.

1. ***Gray-Level Co-ocurence Matrix* (GLCM)**

*Gray-Level Co-occurrence Marix* (GLCM) merupakan ciri statistik orde dua yang merepresentasikan hubungan ketetanggan antar dua piksel dalam sebuah citra keabuan. Metode ini diperkenalkan oleh Haralick pada tahun 1973 [3]. Untuk memperoleh matriks kookurensi dibutuhkan dua komponen yaitu arah dan jarak, dimana arah dinyatakan dalam sudut 0, 45, 90, 135 dan seterusnya, sedangkan jarak dinyatakan dalam jumlah piksel, misalnya 1,2,3, dan seterusnya. Arah ketetanggaan antar dua piksel ditunjukkan pada Gambar 1.



Gambar 1. Arah ketetanggaan antar dua piksel

8 arah ketetanggan antara dua piksel dapat direduksi menjadi 4 arah, yaitu 0, 45, 90, dan 135. Arah 0, 45, 90, dan 135 merupakan *transpose* dari arah 180, 225, 270, dan 315 [3]. Misalkan untuk mendapatkan matriks *framework* pada arah 0 dan 180, tambahkan matriks *framework* 0 dengan matrik *transpose*-nya.

Untuk memperoleh ciri atau fitur statistik GLCM, langkah awal yang dilakukan adalah membentuk matriks kookurensi berdasarkan arah dan jarak yang telah ditentukan sebelumnya. Selanjutnya matriks kookurensi ditambahkan dengan matriks *transpose*-nya agar menjadi simetris. Setelah itu matriks simetris dinormalisasi ke bentuk probabilitas dengan cara membagi masing-masing nilai kookurensi dengan jumlah semua nilai kookurensi. Hasil dari normalisasi matriks ini digunakan untuk menghitung fitur statistik GLCM. Beberapa ciri atau fitur statistik yang akan digunakan dalam penelitian ini adalah sebagai berikut.

|  |  |
| --- | --- |
|  | (1) |

*Energy* atau *Angular Second Moment* (ASM) digunakan untuk mengukur homogenitas sebuah citra.

Dimana menyatakan nilai pada baris dan kolom pada matriks kookurensi.

|  |  |
| --- | --- |
|  | (2) |

*Correlation* digunakan untuk menghitung keterkaitan piksel yang memiliki level keabuan dengan piksel yang memiliki level keabuan .

|  |  |
| --- | --- |
|  | (3) |

*Homogenity* atau *Inverse Different Moment* (IDM) digunakan untuk mengukur homogenitas citra dengan level keabuan sejenis.

|  |  |
| --- | --- |
|  | (4) |

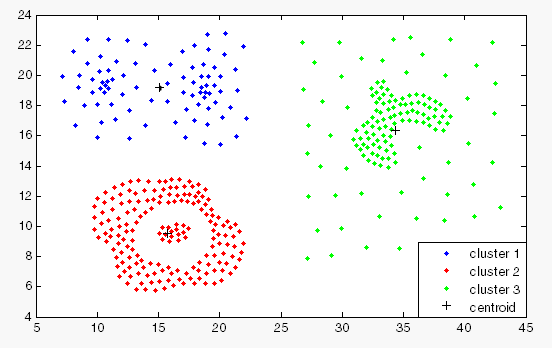
*Entropy* digunakan untuk menghitung level ketidakteraturan bentuk.

|  |  |
| --- | --- |
|  | (5) |

*Contrast* digunakan untuk mengukur variasi pasangan tingkat keabuan dalam sebuah citra.

1. ***K-Means***

Metode *K-Means* dapat diterapkan pada kasus dengan jumlah objek yang sangat besar [4]. *K-Means* mempartisi data ke dalam *cluster* sehingga data yang memiliki karakteristik yang sama dikelompokkan ke dalam satu *cluster* yang sama dan data yang mempunyai karakteristik yang berbeda dikelompokkan ke dalam *cluster* yang lain. Adapun tujuan dari data *clustering* ini adalah untuk meminimalkan *objective* *function* yang diset dalam proses *clustering*, yang pada umumnya berusaha meminimalkan variasi di dalam suatu *cluster* dan memaksimalkan variasi antar *cluster* [5].



Gambar 2. Clustering K-Means

Jika diberikan sekumpulan objek maka algoritma *K-Means* akan mempartisi kedalam buah cluster. Setiap *cluster* memiliki *centroid* dari objek-objek dalam *cluster* tersebut. Pada tahap awal algoritma *K-Means* memilih secara acak buah objek sebagai *centroid*. Selanjutnya jarak antara objek dengan *centroid* dihitung menggunakan fungsi kemiripan. Objek akan ditempatkan dalam *cluster* yang terdekat dihitung dari titik tengah *cluster* atau *centroid*. *Centroid* baru ditetapkan jika semua objek sudah ditempatkan dalam *cluster* terdekat. Proses penentuan *centroid* dan penempatan objek dalam *cluster* diulangi sampai nilai *centroid* konvergen (*centroid* dari semua *cluster* tidak berubah lagi).

1. ***Canberra Distance***

*Feature extraction* merupakan proses yang dilakukan untuk mendapatkan fitur sebuah citra. Fitur tersebut digunakan untuk mengetahui kemiripan antara citra satu dengan yang lain. Proses inilah yang dinamakan proses penghitungan kemiripan (*similarity measure*). Proses perhitungan kemiripan dapat dilakukan dengan menghitung atau mencari jarak antara fitur-fitur yang telah didapat. *Canberra distance* merupakan salah satu metode yang dapat digunakan untuk mencari jarak dari dua objek berdasarkan fitur yang ada. Prosesnya adalah mencari selisih dari fitur objek pertama dan kedua kemudian dibagi dengan jumlah fitur objek pertama dan kedua. Dan seterusnya dilakukan penjumlahan dengan fitur kedua hingga fitur ke- [6].

|  |  |
| --- | --- |
|  | (6) |

Dalam persamaan *Canberra Distance*, pembilang merepresentasikan perbedaan sedangkan penyebut menormalisasikan perbedaan. Jadi, nilai jarak tidak akan pernah melebihi 1. Nilai jarak akan menjadi 1 ketika salah satu dari attribut bernilai 0 [7]. Sehingga hal ini dapat mengantisipasi jika fitur yang didapatkan memiliki rentang nilai yang jauh. Misalkan fitur pertama dan kedua bernilai puluhan, sedangkan fitur ketiga dan keempat bernilai ribuan.

1. ***Recall* dan *Precission***

*Recall* dan *Precision* merupakan ukuran evaluasi yang digunakan untuk mengevaluasi efektivitas dari aplikasi temu kembali citra. *Recall* adalah ukuran dari kemampuan sebuah sistem untuk menampilkan seluruh citra yang relevan. Sedangkan *precision* adalah ukuran dari kemampuan sebuah sistem untuk menampilkan sebagian citra yang relevan [8].

Misalnya sebuah citra *query* yang akan dicari dari kumpulan citra (*Collection Images*). Himpunan menyatakan kumpulan citra yang relevan. adalah jumlah citra yang relevan. Anggaplah sebuah sistem pencarian menghasilkan himpunan citra hasil temu-kembali (*Image Results*), . Kumpulan citra di dalam himpunan diurutkan berdasarkan derajat relevansinya dengan citra *query* . *Ranking* yang dihasilkan dimulai dari derajat relevansi tertinggi. adalah jumlah elemen dari himpunan citra yang dihasilkan sistem hasil . adalah jumlah citra yang merupakan irisan himpunan dan himpunan . Hubungan antara himpunan-himpunan ini diperlihatkan pada gambar berikut [9].



Gambar 3. Diagram himpunan citra

Untuk menghitung *recall* dapat ditentukan dengan persamaan (7) dan untuk menghitung *precision* dapat ditentukan dengan persamaan (8)

|  |  |
| --- | --- |
|  | (7) |
|  | (8) |

Pada dasarnya, nilai *recall* dan *precision* berada pada rentang 0 sampai 1 atau dapat juga dinyatakan dalam bentuk prosentase. Oleh karena itu, suatu sistem temu kembali yang baik adalah yang mampu memberikan nilai *recall* dan *precision* mendekati 1 atau 100%.

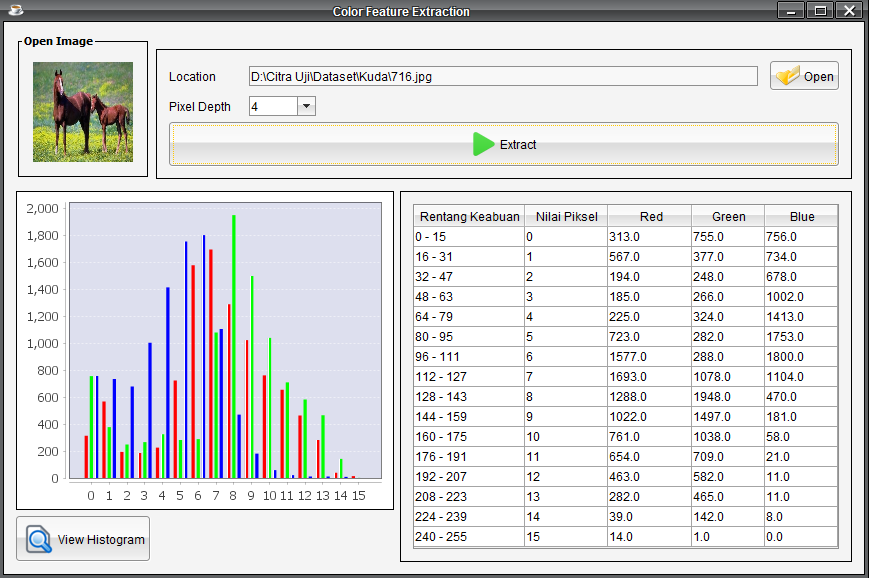
1. **RANCANGAN SISTEM DAN APLIKASI**



Gambar 4. Gambaran umum sistem.

1. **Ekstraksi Fitur Warna**

Ektraksi fitur warna dibuat dalam bentuk histogram pada masing-masing ruang warna (RGB). Setiap ruang warna akan dikuantisasi dengan kedalaman piksel 4 bit. Artinya setiap ruang warna akan dibuat histogram frekuensinya kedalam 16 bin (antara 0 – 15). Proses ini akan menghasilkan vektor fitur warna dengan panjang 3x16 elemen. Nilai tiap elemen dalam vektor merupakan nilai frekuensi piksel dalam range yang merupakan hasil pembagian dari nilai 256 dibagi ke dalam 16 bin untuk setiap ruang warna (RGB).

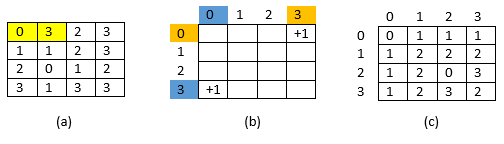


Gambar 5. Ekstraksi fitur warna

1. **Ekstraksi Fitur Tekstur**

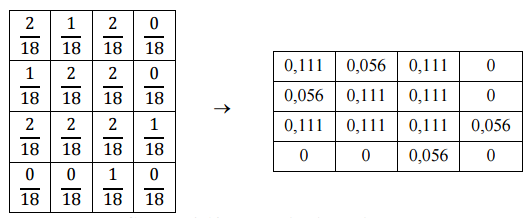
Ekstraksi fitur tekstur dilakukan dengan menggunakan metode *Gray-Level Co-occurrence Matrix* (GLCM). Arah yang digunakan dalam proses ekstraksi ini yaitu 0, 45, 90, dan 135. Kemudian jarak ketetanggaan antar piksel yang digunakan yaitu 1.

Proses yang pertama kali dilakukan dalam metode GLCM adalah mengkonversi citra RGB menjadi citra *grayscale*. Selanjutnya adalah membentuk matriks simetris yang diperoleh dari penjumlahan matriks kookurensi dan *transpose* matriks kookurensi.



Gambar 6. Ilustrasi pembentukan matriks simetris

Setelah matriks simetris terbentuk, selanjutnya adalah menormalisasi matriks dengan cara membagi masing-masing nilai kookurensi dengan jumlah semua nilai kookurensi yang ada pada matriks.



Gambar 7. Ilustrasi normalisasi matriks

Hasil dari normalisasi matriks ini digunakan untuk menghitung fitur statistik *energy*, *correlation*, *homogenity*, *entropy*, dan *contrast*. Proses ini akan menghasilkan vektor fitur tekstur dengan panjang 4x5 elemen. Nilai tiap elemen merupakan hasil perhitungan fitur statistik untuk arah 0, 45, 90, dan 135.

1. ***Clustering***

*Clustering* dilakukan untuk mengelompokkan citra *database* berdasarkan kemiripan fitur warna dan tekstur. Algoritma *clustering* yang digunakan dalam penelitian ini yaitu algoritma *K-Means*. Tujuan dilakukannya *clustering* adalah mempercepat waktu komputasi serta meningkatkan keakuratan hasil pencarian.

Proses *clustering* akan menghasilkan nilai *centroid* atau koordinat pusat masing-masing *cluster*. Nilai *centroid* ini yang dijadikan acuan dalam pengukuran kemiripan dengan citra *query*. Hasil pengukuran kemiripan yang memiliki jarak terdekat akan dijadikan *cluster* kandidat. Dengan adanya *clustering*, proses perhitungan kemiripan tidak dilakukan pada keseluruhan citra dalam *database* melainkan hanya pada citra yang berada pada *cluster* kandidat. Berikut ini adalah ilustrasi perbandingan perhitungan kemiripan menggunakan *clustering* dengan perhitungan kemiripan tanpa menggunakan *clustering*.



Gambar 8. Ilustrasi perhitungan kemiripan  
(a) Menggunakan clustering (b) Tidak menggunakan clustering

1. **Perhitungan Kemiripan**

Kemiripan antara citra *query* dengan citra *database* diukur dari dua jenis karakteristik fitur yaitu fitur warna dan fitur tekstur. Perhitungan kemiripan dilakukan dengan menggunakan metode *Canberra Distance*. Karena masing-masing fitur mewakili aspek properti pada sebuah citra, maka diperlukan bobot yang tepat untuk menggabungkan kedua fitur tersebut. Nilai bobot memiliki rentang antara 0 dan 1. Perhitungan untuk mendapatkan citra kembalian berdasarkan bobot adalah sebagai berikut :

|  |  |
| --- | --- |
|  | (9) |

Dimana adalah nilai hasil pembobotan, adalah bobot untuk fitur warna, adalah hasil perhitungan *Canberra* pada fitur warna, adalah bobot untuk fitur tekstur dan adalah hasil perhitungan *Canberra* pada fitur tekstur. Citra hasil kembalian akan diurut berdasarkan nilai yang terkecil.

1. **HASIL DAN PEMBAHASAN**
2. **Pengujian Nilai Bobot**

Pengujian ini dilakukan untuk menguji peran bobot dan menetapkan bobot terbaik untuk masing-masing fitur. Citra *query* yang digunakan dalam pengujian ini sebanyak lima buah citra untuk setiap kelompok. Kemudian hasil temu-kembali (pencarian) yang digunakan yaitu sebanyak 20 citra. Berikut ini merupakan hasil pengujian yang telah dilakukan.

Tabel 1. Hasil percobaan bobot warna = 0,05; bobot tekstur = 0,95

|  |  |  |  |
| --- | --- | --- | --- |
| Bobot warna | Bobot tekstur | Kategori | Total citra relevan |
| 0.05 | 0.95 | Penduduk | 60 |
| 0.05 | 0.95 | Bangunan | 63 |
| 0.05 | 0.95 | Bis | 76 |
| 0.05 | 0.95 | Bunga | 93 |
| 0.05 | 0.95 | Dinosaurus | 100 |
| 0.05 | 0.95 | Gajah | 38 |
| 0.05 | 0.95 | Gunung | 46 |
| 0.05 | 0.95 | Kuda | 89 |
| 0.05 | 0.95 | Makanan | 38 |
| 0.05 | 0.95 | Pantai | 60 |
| Rata-rata Presisi | | | 66.30% |

Tabel 2. Hasil percobaan bobot warna = 0,5; bobot tekstur = 0,5

|  |  |  |  |
| --- | --- | --- | --- |
| Bobot warna | Bobot tekstur | Kategori | Total citra relevan |
| 0.5 | 0.5 | Penduduk | 88 |
| 0.5 | 0.5 | Bangunan | 68 |
| 0.5 | 0.5 | Bis | 81 |
| 0.5 | 0.5 | Bunga | 94 |
| 0.5 | 0.5 | Dinosaurus | 100 |
| 0.5 | 0.5 | Gajah | 86 |
| 0.5 | 0.5 | Gunung | 47 |
| 0.5 | 0.5 | Kuda | 98 |
| 0.5 | 0.5 | Makanan | 80 |
| 0.5 | 0.5 | Pantai | 76 |
| Rata-rata Presisi | | | 81.80% |

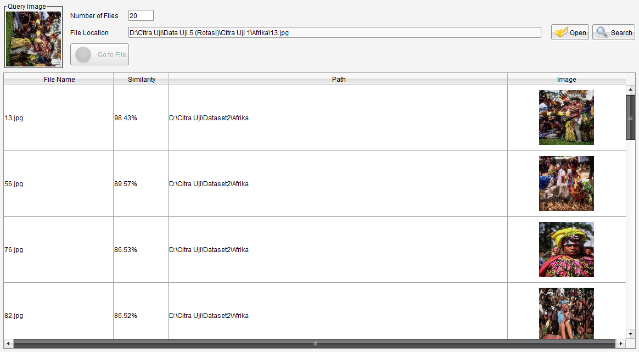
Tabel 3. Hasil percobaan pertama bobot warna = 0,95; bobot tekstur = 0,5

|  |  |  |  |
| --- | --- | --- | --- |
| Bobot warna | Bobot tekstur | Kategori | Total citra relevan |
| 0.95 | 0.5 | Penduduk | 92 |
| 0.95 | 0.5 | Bangunan | 67 |
| 0.95 | 0.5 | Bis | 85 |
| 0.95 | 0.5 | Bunga | 93 |
| 0.95 | 0.5 | Dinosaurus | 100 |
| 0.95 | 0.5 | Gajah | 89 |
| 0.95 | 0.5 | Gunung | 47 |
| 0.95 | 0.5 | Kuda | 100 |
| 0.95 | 0.5 | Makanan | 84 |
| 0.95 | 0.5 | Pantai | 73 |
| Rata-rata Presisi | | | 83% |

Dari hasil percobaan di atas dapat dilihat bahwa bobot sangat mempengaruhi tingkat presisi pencarian citra. Bila dilihat pada tabel 1, 2, dan 3, rata-rata nilai presisi terbesar diperoleh ketika bobot yang digunakan adalah 0.95 untuk fitur warna dan 0,5 untuk fitur tekstur. Bobot yang digunakan pada fitur warna lebih besar dibandingkan dengan fitur tekstur, artinya kelompok citra yang digunakan memiliki fitur dominan terhadap warna.

1. **Pengujian Citra Rotasi**

Pengujian ini dilakukan untuk menguji apakah sistem temu kembali citra dapat menemukan citra yang tepat apabila citra telah dirotasi dengan sudut 90o.

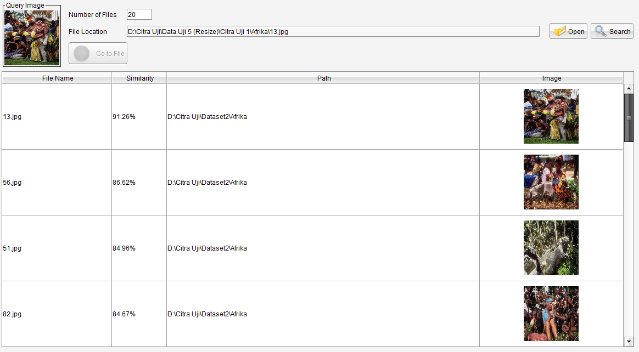
****

Gambar 9. Hasil pengujian citra rotasi

Gambar di atas memperlihatkan bahwa sistem mampu menemukan citra yang tepat dengan tingkat kemiripan 98,43% terhadap citra yang telah dirotasi 90o. Hal ini menunjukkan bahwa fitur yang digunakan tidak terlalu berpengaruh oleh rotasi karena sistem masih mampu menemukan citra yang tepat pada urutan pertama.

1. **Pengujian Citra *Resize***

Pengujian ini dilakukan untuk menguji apakah sistem temu kembali citra dapat menemukan citra yang tepat apabila citra telah diubah ukurannya menjadi lebih kecil, yaitu 50% dari ukuran aslinya.

****

Gambar 10. Hasil pengujian citra resize

Gambar di atas memperlihatkan bahwa sistem mampu menemukan citra yang tepat dengan tingkat kemiripan 91,26% terhadap citra yang telah diubah ukurannya menjadi lebih kecil 50% dari ukuran aslinya. Hal ini menunjukkan bahwa fitur yang digunakan tidak terlalu berpengaruh oleh perubahan ukuran karena sistem masih mampu menemukan citra yang tepat pada urutan pertama.

1. **Evaluasi Tingkat Presisi**

Evaluasi ini dilihat berdasarkan lima pengujian yang telah dilakukan. Masing-masing pengujian diambil rata-rata presisinya, kemudian dari kelima pengujian ini diambil nilai tengahnya untuk mendapatkan tingkat presisi aplikasi temu kembali citra.

Tabel 4. Evaluasi tingkat presisi

|  |  |
| --- | --- |
| Pengujian | Rata-rata presisi (%) |
| Pengujian Citra Asli | 83 |
| Pengujian Citra Rotasi | 83 |
| Pengujian Citra Resize | 78.3 |
| Pengujian Citra Asing | 78.7 |
| Pengujian Citra Lapangan | 84 |
| Mean | 81.4 |

Tabel di atas menunjukkan hasil dari lima pengujian yang telah dilakukan. Dari tabel pengujian tersebut didapatkan rata-rata persentase presisinya adalah 81,4%.

1. **Evaluasi Performa Sistem**

Evaluasi ini dilihat berdasarkan pengujian perhitungan nilai *recall* dan presisi yang telah dilakukan. Hasil dari pengujian ini adalah nilai rata-rata presisi pada setiap citra uji untuk *recall* 10% hingga 100%. Selanjutnya, akan diambil nilai tengah dari kesuluruhan citra uji untuk setiap rentang *recall*.

Tabel 5. Evaluasi performa sistem

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Recall (%) | Rata-rata presisi (%) | | | | | Mean (%) |
| Citra uji 1 | Citra uji 2 | Citra uji 3 | Citra uji 4 | Citra uji 5 |
| 10 | 90.83 | 82.42 | 87.12 | 92.7 | 86.64 | 87.94 |
| 20 | 74.33 | 74.24 | 81.16 | 78.77 | 82.82 | 78.26 |
| 30 | 64.3 | 61.82 | 68.17 | 69.42 | 73.99 | 67.54 |
| 40 | 53 | 47.43 | 59.76 | 60.62 | 64.14 | 56.99 |
| 50 | 34.36 | 36.42 | 48.75 | 49.83 | 54.14 | 44.7 |
| 60 | 31.04 | 20.38 | 37.93 | 36.29 | 40.8 | 33.29 |
| 70 | 19.94 | 15.83 | 27.24 | 19.99 | 28.77 | 22.35 |
| 80 | 10 | 10 | 16.06 | 10 | 18.33 | 12.88 |
| 90 | 10 | 10 | 9.89 | 9.89 | 16.87 | 11.33 |
| 100 | 9.9 | 9.01 | 9.9 | 9.901 | 10 | 9.742 |

Tabel di atas menunjukkan terjadi penurunan nilai rata-rata presisi ketika nilai *recall* semakin besar. Hal ini terjadi karena terdapat kelompok yang tidak memiliki relevansi antar citra yang tinggi, sehingga sangat mempengaruhi tingkat presisi ketika *recall* semakin besar (atau mendekati 100%). Tingkat relevansi antar citra ini dapat dilihat dari komposisi warna dan tekstur. Misalnya saja untuk kelompok citra gunung, kelompok ini tidak memiliki komposisi warna dan tekstur yang sama dimana terdapat berbagai jenis citra seperti pegunungan bersalju, pegunungan hijau, pegunungan di sekitar danau, dan lain sebagainya. Hal ini menyebabkan tingkat presisi menjadi kecil karena sistem temu-kembali pada penelitian ini sulit mengenali pola yang acak.

1. **Evaluasi Performa *Clustering***

Evaluasi ini dilihat berdasarkan pengujian terhadap waktu proses *clustering* dan waktu proses perhitungan kemiripan. Kemudian dihitung rata-rata presisi yang didapatkan ketika menggunakan *clustering*.

Tabel 6. Evaluasi performa clustering

|  |  |  |  |
| --- | --- | --- | --- |
| Percobaan | Waktu clustering (s) | Rata-rata waktu perhitungan kemiripan (s) | Rata-rata presisi (%) |
| Tanpa Clustering | 0 | 0.1045 | 81.5 |
| Jumlah Cluster 2 | 2.878 | 0.0629 | 81.5 |
| Jumlah Cluster 10 | 11.227 | 0.0116 | 76.5 |
| Jumlah Cluster 20 | 22.652 | 0.0078 | 81 |

Tabel di atas memperlihatkan bahwa ketika jumlah *cluster* semakin besar, proses *clustering* membutuhkan waktu lebih lama. Sebaliknya waktu proses perhitungan kemiripan akan lebih cepat ketika jumlah *cluster* semakin besar. Sementara itu untuk tingkat presisi yang diperoleh bisa lebih baik atau lebih buruk karena karakteristik dari *clustering K-Means* bersifat unik (selalu berubah-ubah). Hal ini dikarenakan pembangkitan *centroid* awal dilakukan secara random.

1. **KESIMPULAN**

Berdasarkan hasil implementasi dan pengujian pada bab sebelumnya, maka dapat diambil beberapa kesimpulan sebagai berikut :

1. Aplikasi temu-kembali citra yang dibuat berhasil melakukan pencarian berdasarkan tingkat kemiripan antara antara citra *query* dengan citra *database* dengan rata-rata presisi yaitu 81,4%.
2. Penentuan nilai bobot sangat berpengaruh terhadap hasil pencarian. Untuk itu diperlukan bobot yang tepat sehingga mendapatkan hasil yang maksimal.
3. Penggunaan fitur warna dan tekstur menggunakan metode histogram dan GLCM tidak terpengaruh oleh rotasi dan perubahan ukuran untuk citra yang sama.
4. *Clustering* *K-Means* dapat mempercepat waktu proses perhitungan kemiripan, karena proses perhitungan kemiripan hanya dilakukan terhadap *cluster* kandidat. Sementara itu, untuk tingkat presisi yang diperoleh bisa lebih baik atau lebih buruk karena pembangkitan *centroid* awal yang dilakukan secara random.

**DAFTAR PUSTAKA**

[1] C. Kavitha, D. Rao, dan D. Govardhan, “Image Retrieval Based On Color And Texture Features Of The Image Sub-Blocks,” *Int. J. Comput. Appl.*, vol. 15, no. 7, pp. 33–37, 2011.

[2] Z. Zarcelonia, “Perancangan Aplikasi Perbesaran Citra Dengan Metode Proyeksi Cahaya,” 2013.

[3] A. Febriani, “Identifikasi Diabetic Retinopathy Melalui Citra Retina Menggunakan Modified K-Nearest Neighbor,” Universitas Sumatera Utara, 2015.

[4] R. Nainggolan, “Algoritma Modified K-MEANS Clustering Pada Penentuan Cluster Centre Berbasis Sum Of Squared Error (SSE),” Universitas Sumatera Utara, 2014.

[5] Y. Agusta, “K-Means – Penerapan, Permasalahan dan Metode Terkait,” *J. Sist. dan Inform.*, vol. 3, no. Pebruari, pp. 47–60, 2007.

[6] R. A. Tama, N. Suciati, dan A. Y. Wijaya, “Implementasi Temu Kembali Citra Berbasis Isi Dengan Fitur Titik-Titik Signifikan,” pp. 1–6, 2011.

[7] M. Jamaluddin, N. Suciati, dan A. Y. Wiajaya, “Implementasi Temu Kembali Citra Tekstur Menggunakan Rotated Wavelet Filter,” 2011.

[8] Kondekar, Kolkure, dan Kore, “Image Retrieval Techniques based on Image Features : A State of Art approach for CBIR,” *Int. J. Comput. Sci. Inf. Secur.*, vol. 7, no. 1, pp. 69–76, 2010.

[9] H. Siagian, “Temu Kembali Citra Wajah Berdasarkan Pengukuran Kemiripan Fitur Dengan Menggunakan Jaringan Bayesin,” Universitas Sumatera Utara, 2013.