

Pengklasteran K-Means Database Citra Untuk Meningkatkan Akurasi Pencarian Query CBIR Menggunakan Intensitas Warna

Juli Rejito

Program Studi Teknik Informatika, Jurusan Matematika,
Fakultas MIPA, Universitas Padjadjaran
Jl. Raya Bandung Sumedang Km 21 Jatinangor Sumedang 45363
e-mail: jrejito@yahoo.com

ABSTRAK

Meningkatnya jumlah gambar digital yang tersimpan dalam media penyimpanan database dan kemampuan komputer untuk menyediakan kebutuhan informasi pengguna dengan cepat telah menjadi kebutuhan penting saat ini. Efisiensi pencarian citra dalam database berukuran besar menggunakan sistem CBIR (*Content Based Image Retrieval*) telah menjadi kebutuhan. Makalah ini bertujuan untuk mendapatkan solusi peningkatan akurasi query CBIR pada database WANG sebanyak 1.000 record database citra yang memiliki resolusi 256 x 384 piksel dan 384 x 256 piksel. Solusi yang diusulkan berupa model arsitektur sistem yang mengintegrasikan optimasi query dan pengklasteran partisi dengan harapan bahwa model ini akan meningkatkan tingkat akurasi pencarian dan efisiensi waktu hasil pencarian. Pembentukan kluster didasarkan pada nilai PSNR minimum dan maksimum yang dihasilkan dari perbandingan antara ekstraksi fitur warna record database citra dan citra dasar, yang ditunjukkan dalam pembentukan 2, 4, 8, 16, dan 32 cluster sebagai indeks cluster sekaligus difungsikan sebagai penyaring. Hasil implementasi model ini menunjukkan bahwa akurasi nilai F-Score untuk query CBIR non klaster meningkat ketika diterapkan query CBIR menggunakan 5 klaster menggunakan ekstraksi fitur intensitas warna yang ditunjukkan oleh nilai F-Score dari 0,14 menjadi 0,17.

Kata kunci : CBIR, Pengklasteran, PSNR, WANG Database, Intensitas Warna, F-Score

ABSTRACT

The increasing number of digital images stored on a computer database storage and the ability that provide users need to get information quickly has become an important requirement at this time . The efficiency of image retrieval in large databases using a CBIR system (Content Based Image Retrieval) has become a necessary . This paper aims to obtain solutions to improve the accuracy of CBIR query on a database of 1,000 records WANG image that has a resolution of 256 x 384 pixels and 384 x 256 pixels . The proposed solution in the form of a model that integrates system architecture and query optimization clustering partition in the hope that this model will improve the accuracy and efficiency time of search results . The formation of clusters is based on the minimum and maximum PSNR values resulting from the comparison between the color feature extraction and image database records basic image , shown in the formation of 2 , 4 , 8 , 16 , and 32 clusters as well as cluster index functioned and also as a filter . The results of the implementation of this model shows that the accuracy of the F - Score values for non CBIR query clusters increases when the query is applied CBIR using feature extraction 5 clusters using color intensity indicated by the value of the F - Score from 0.14 becomes 0.17 .

Keywords : CBIR, Clustering, PSNR, WANG Database, Color Histogram, F-Score

1. Pendahuluan

Pencarian citra di internet saat ini sudah banyak tersedia dalam bentuk mesin pencari citra seperti *Google Image Search*, *Flickr*, dan *AltaVista Image*

Search. Metoda pencarian citra dibagi dalam dua kategori yaitu *text based image retrieval* (TBIR) dan *content based image retrieval* (CBIR). Teknik pencarian menggunakan kategori TBIR dikembangkan oleh beberapa peneliti antara lain penelitian TBIR menggunakan model pembelajaran kategori [1], metoda *probabilistic Latent Semantic Analysis* (pLSA) [2], TBIR dengan pengelompokan spesifik objek dari hasil pencarian web [3]. Google sendiri saat ini sudah mengembangkan sistem pencarian berdasarkan kemiripan citra (CBIR) dengan cara memasukkan query citra. Teknik pencarian CBIR ditindaklanjuti oleh peneliti lain diantaranya penyajian kemiripan citra dalam struktur graf dan pembentukan subset koleksi citra untuk mempercepat pencarian [4], pengukuran kemiripan visual yang menggabungkan relevansi citra dan kinerja pencarian [5], ide dengan *co-reranking* untuk pencarian citra [6], dan pengenalan *crowd reranking* diambil dari karakteristik hasil *visual pattern* dengan mengambil dari hasil pencarian citra dengan menggunakan beberapa model pencarian citra [7].

Sistem CBIR merupakan salah satu bidang riset yang sedang berkembang pesat dan banyak penelitian masih dilakukan pada saat ini. Pencarian citra menggunakan fitur visual telah banyak dilakukan, demikian juga sistem CBIR telah banyak dikembangkan, namun demikian dalam menerapkan sistem CBIR tersebut masih banyak mengalami kendala. Masalah ini muncul karena adanya pemisah yang bersifat tingkat rendah (*low level*) di satu sisi dan tingkat tinggi (*high level*) pada sisi lain yang sulit diselesaikan dalam sistem CBIR [8]. Pemisah ini muncul akibat dari proses ekstraksi fitur citra, pengindeksan, dan query dalam sistem CBIR akan diinterpretasikan secara berbeda oleh pengguna yang melakukan pencarian citra dalam fitur tingkat tinggi yang disebut sebagai kesenjangan semantik (lihat [9]). Kesenjangan semantik adalah perbedaan interpretasi sebuah citra dari sudut pandang pengguna (bersifat *high level*) dengan sudut pandang sistem (bersifat *low level*), dimana hasil pencarian citra diperoleh dari pengolahan citra piksel demi piksel, sehingga hasil pencariannya kurang mewakili maksud dari pengguna. Tingkat kemiripan pencarian citra pada sistem CBIR tergantung pada jarak mutlak dari fitur citra terhadap kemiripan citra yang dicari [10]. Disamping itu, untuk melakukan ekstraksi fitur citra memerlukan banyak waktu apalagi kalau jumlah citranya sangat banyak yang akan mengakibatkan lambatnya perolehan kembali citra yang dicari (lihat [11]).

Pengukuran kemiripan citra adalah bagian terpenting dalam sistem CBIR, baik untuk citra asli maupun citra yang terkompresi. Ketika melakukan evaluasi terhadap citra yang terkompresi, maka tingkat kemiripan citra yang mendekati sistem visual manusia (*human visual system*) akan mendapatkan citra yang lebih akurat. Pengukuran yang sering dipergunakan dalam pencarian kemiripan citra adalah dengan *mean squared error* (MSE) dan *peak signal-to-noise ratio* (PSNR). Nilai MSE dan PSNR akan menunjukkan tingkat kritikal kemiripan citra terhadap penyimpangan citra yang dibandingkan (lihat [12] dan [13]). Penyimpangan nilai tersebut pada citra digital akan mengakibatkan

terjadinya penurunan kualitas citra, sehingga pada akhirnya akan mempengaruhi penilaian dari sistem visual manusia. Keuntungan pengukuran kualitas citra menggunakan nilai MSE dan *PSNR* adalah perhitungan yang sangat cepat dan mudah untuk diimplementasikan [14].

Beberapa kajian yang dilakukan oleh peneliti sebelumnya masih memunculkan beberapa masalah yang harus diselesaikan antara lain

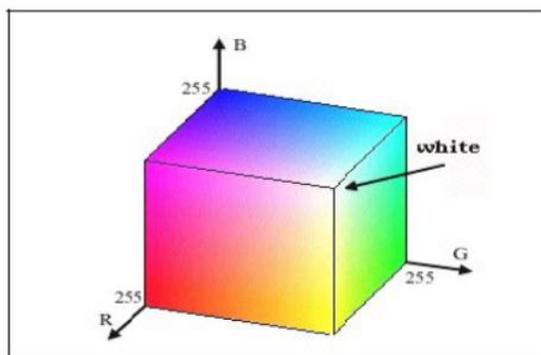
1. Adanya kesenjangan semantik dalam sistem *CBIR* antara tingkat rendah dan pengguna yang akan melakukan pencarian dalam bentuk fitur tingkat tinggi sehingga sistem *CBIR* memerlukan waktu yang cukup lama pada proses pencarian citra terlebih apabila jumlah citra yang dicari sangat banyak.
2. Pencarian konten citra untuk database citra berukuran besar pada sistem *CBIR* masih mengalami kendala karena ukuran database yang besar mengakibatkan pencarian citra menjadi lama.
3. Diperlukan alternatif yang lain yang mampu meningkatkan tingkat akurasi dan mampu mereduksi waktu proses pencarian konten citra ke dalam database citra terlebih untuk database citra berukuran besar sehingga pencarian konten citra diharapkan dapat lebih singkat dan dapat menjembatani kesenjangan semantik.

2. Tinjauan Pustaka

2.1 Ruang Warna

Warna pada dasarnya adalah hasil persepsi cahaya dalam spektrum wilayah yang terlihat oleh retina mata, dan memiliki panjang gelombang antara 400 nm sampai dengan 700 nm. Ruang warna RGB dapat divisualisasikan sebagai sebuah kubus seperti Gambar 1, dengan tiga sumbunya yang mewakili komponen warna R (*red*) atau merah, G (*green*) atau hijau, B (*blue*) atau biru. Salah satu pojok alasnya yang berlawanan menyatakan warna hitam ketika nilai $R=G=B=0$, sedangkan pojok atasnya yang berlawanan menyatakan warna putih ketika nilai $R=G=B=255$ (sistem warna 8 bit bagi setiap komponennya).

Penyajian warna didasarkan pada tiga warna dasar di mana setiap warna dapat direproduksi dengan pencampuran satu set yang sesuai tiga warna dasar tadi. Dengan cara ini, penyajian yang kuantitatif tentang warna tertentu dapat ditentukan oleh tiga vektor komponennya di dalam ruang 3 dimensi pada sistem koordinat. Himpunan dari semua warna membentuk suatu ruang vektor disebut *color space* atau *color model*.



Gambar 1. Komponen warna RGB sebagai vektor intensitas warna

Informasi warna diwakili dalam *RGB* (*red, green, blue*) secara luas menggunakan kombinasi merah, hijau dan biru yang dibentuk dalam sistem koordinat kartesius. Warna dapat digambarkan sebagai suatu penyajian ruang metrik warna sehingga suatu perbedaan warna dapat dimengerti yang diwakili oleh jarak *Euclidean*. Bentuk persamaan transformasi seperti ditunjukkan pada persamaan (1) dipergunakan untuk konversi nilai vektor dari RGB ke bentuk $L^*a^*b^*$, nilai masing-masing L^* pada persamaan (2), a^* pada persamaan (3,) dan b^* pada persamaan (4).

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 0.490 & 0.310 & 0.200 \\ 0.177 & 0.813 & 0.011 \\ 0.000 & 0.010 & 0.990 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (1)$$

$$L^* = 25 \left(\frac{100Y}{Y_0} \right)^{\frac{1}{3}} - 16 \quad (2)$$

$$a^* = 500 \left[\left(\frac{X}{X_0} \right)^{\frac{1}{3}} - \left(\frac{Y}{Y_0} \right)^{\frac{1}{3}} \right] \quad (3)$$

$$b^* = 200 \left[\left(\frac{Y}{Y_0} \right)^{\frac{1}{3}} - \left(\frac{Z}{Z_0} \right)^{\frac{1}{3}} \right] \quad (4)$$

Ketiga persamaan tersebut diatas dibatasi untuk $1 \leq 100Y \leq 100$ harus dipenuhi, nilai antara $[XYZ]^T$ adalah nilai tristimulus dari *CIE XYZ*, dan $[X_0Y_0Z_0]^T$ adalah referensi dari warna putih. L^* berhubungan dengan *brightness*, a^* untuk *red-green*, dan b^* untuk *yellow-blue*. Bentuk yang sama dapat ditemukan dalam system koordinat $L^*u^*v^*$.

2.2 Pengukuran Kualitas Citra

Pengukuran kualitas citra terbagi menjadi 2 (dua) kelompok yang didasarkan pada HVS (*Human visual system*) atau sistem visualisasi manusia [15]. Kelompok pertama adalah sistem pengukuran konvensional yang menggunakan teknik perhitungan nilai SNR (*Signal to Noise Ratio*) dan *PSNR* (*Peak Signal to Noise Ratio*). Kelompok yang kedua didasarkan pada kriteria ketelitian perubahan sinyal informasi dan distorsi citra yang didasarkan pada

perhitungan nilai SSIM (*Structural Similarity Index*) dan VroiWQI (*Visual region of interest Weighted Quality Index*).

Pengukuran kualitas citra secara konvensional banyak dipergunakan karena kemudahan dalam perhitungan, memiliki pemahaman secara fisik dan secara matematikpun mudah untuk dipergunakan untuk tujuan optimisasi, walaupun tidak terlalu baik dalam pencocokan kualitas visualnya. Pengukuran ini terdiri dari beberapa formula [16] yaitu *MAE* (*Maximum Absolute Error*), *MSE* (*Mean Squared Error*), *RMSE* (*Root Mean Squared Error*), *SNR* dan *PSNR*.

1. *Maximum Absolute Error (MAE)*

MAE adalah nilai maximum dari harga mutlak (*absolute*) selisih antara citra masukan $f(x, y)$ dan citra keluaran $g(x, y)$. Secara matematis perhitungan *MAE* dituliskan dalam bentuk persamaan (5).

$$MAE = \max |f(x, y) - g(x, y)| \tag{5}$$

di mana $f(x, y)$: Nilai intensitas citra awal/asli pada posisi (x, y)
 $g(x, y)$: Nilai intensitas citra hasil pada posisi (x, y)

2. *Mean Squared Error (MSE)*

MSE adalah nilai rata-rata kuadrat nilai error antara citra masukan $f(x, y)$ dengan citra keluaran $g(x, y)$, di mana kedua citra tersebut memiliki ukuran yang sama. Nilai *MSE* yang baik adalah mendekati nol ($MSE \approx 0$). Secara matematis perhitungan nilai *MSE* menggunakan persamaan (6).

$$MSE = \frac{1}{MN} \sum_{x=1}^M \sum_{y=1}^N [f(x, y) - g(x, y)]^2 \tag{6}$$

di mana M, N : Lebar dan tinggi citra
 $f(x, y)$: Nilai intensitas citra awal/asli pada posisi (x, y)
 $g(x, y)$: Nilai intensitas citra hasil pada posisi (x, y)

3. *Root Mean Squared Error (RMSE)*

RMSE adalah nilai akar kuadrat hasil dari nilai rata-rata kuadrat error. Secara matematis dirumuskan dalam bentuk persamaan (7).

$$RMSE = \sqrt{MSE} \tag{7}$$

4. *Signal to Noise Ratio (SNR)*

Nilai SNR diformulasikan dalam bentuk persamaan (8).

$$SNR = 10 \log_{10} \left\{ \frac{\sum_{x=1}^M \sum_{y=1}^N f(x, y)^2}{\sum_{x=1}^M \sum_{y=1}^N [f(x, y) - g(x, y)]^2} \right\} \tag{8}$$

5. *Peak Signal to Noise Ratio (PSNR)*

PSNR merupakan nilai perbandingan antara nilai maksimum citra hasil rekonstruksi dengan nilai akar kuadrat dari *MSE* atau sama dengan nilai *RMSE*. Untuk citra 8-bit nilai piksel maksimum adalah 255. Kriteria kualitas gambar akan semakin bagus jika hasil *PSNR* semakin besar dan secara matematis nilai *PSNR* dihasilkan dari nilai *MSE* yang ditunjukkan pada persamaan (9).

$$PSNR = 10 \log_{10} \left[\frac{255^2}{MSE} \right] dB \tag{9}$$

2.3 Pengklasteran Menggunakan *K-Means*

K-Means merupakan sebuah algoritma untuk mengelompokkan sejumlah obyek berdasarkan atribut-atribut menjadi *k* buah kelompok. *K* disini adalah nilai *integer* positif. Pengelompokkan dilakukan dengan meminimasi jarak antara data dan pusat klaster yang berkorespondensi. Asumsikan bahwa *n* adalah obyek pengamatan dan *p* adalah pengukuran peubah. Didefinisikan $X(i, j)$ merupakan nilai dari obyek ke-*i* ($i = 1, 2, \dots, k$) pada peubah ke-*j* ($j = 1, 2, \dots, p$).

Diasumsikan bahwa pengukuran jarak yang akan digunakan yaitu *euclidean distance*. Ambil $P(n, K)$ sebagai partisi yang menunjukkan bahwa setiap *n* obyek dimasukkan pada salah satu klaster $1, 2, \dots, k$. Rata-rata (*mean*) dari peubah ke-*j* pada klaster ke-*l* akan ditunjukkan oleh $\bar{X}(l, j)$ dan jumlah obyek yang terdapat pada klaster ke-*l* akan ditunjukkan oleh $n(l)$. Maka, jarak antara obyek ke-*i* dan klaster ke-*l* dapat dihitung dengan menggunakan persamaan (10).

$$D(i, l) = \left(\sum_{j=1}^p [X(i, j) - \bar{X}(l, j)]^2 \right)^{1/2} \tag{10}$$

Didefinisikan $[P(n, K)] = \sum_{i=1}^n D[i, l(i)]^2$, merupakan komponen *error* dari partisi, di mana $l(i)$ adalah klaster yang beranggotakan obyek ke-*i*, dan $D[i, l(i)]$ adalah jarak *Euclidean* antara obyek *i* dan *mean* klaster dari klaster yang beranggotakan obyek tersebut. Prosedur dari pengelompokkan ini yaitu mencari sebuah partisi dengan komponen *error* yang kecil dengan memindahkan obyek dari satu klaster ke klaster lainnya sampai tidak terjadi perpindahan obyek tersebut sehingga tidak terjadi pengurangan nilai *E*.

Penentuan nilai awal *K* yang digunakan sebagai estimasi awal dari pusat klaster dapat menggunakan beberapa metoda yaitu :

1. Pilih objek-objek *K* yang pertama pada sampel sebagai vektor rata-rata klaster *K* yang awal.
2. Pilih objek-objek *K* yang saling terpisah sangat jauh.
3. Mulai dengan sebuah nilai percobaan untuk *K* yang lebih besar dari yang dianggap perlu, dan memulai untuk membentuk jarak pusat klaster di interval dari standar deviasi pada tiap peubah.
4. Pilih *K* dan konfigurasi nilai awal klaster berdasarkan penjelasan sebelumnya.

Berikut merupakan langkah-langkah dalam pembentukan klaster berdasarkan algoritma *K-Means*:

1. Jika didefinisikan jumlah dari elemen baris adalah $Sum(i)$, maka klaster awal dapat dibentuk dengan mempertimbangkan kasus *i* sebagai bagian dari klaster ke-*j*, di mana *j* adalah pembulatan dari $\frac{K[Sum(i) - Min]}{Max - Min} + 1$, di mana *Max* dan *Min* adalah nilai maksimum dan minimum dari $Sum(i)$.

2. Menghitung $B(i, j)$, yaitu rata-rata peubah ke- j seluruh kasus pada klaster ke- i .
3. Jarak antara kasus ke- i dan klaster ke- l didefinisikan dalam persamaan (11).

$$D(i, l) = \left(\sum_{j=1}^p [X(i, j) - \bar{X}(l, j)]^2 \right)^{1/2} \quad (11)$$

Error pada partisi tertentu didefinisikan pada persamaan (12).

$$E [P(n, K)] = \sum_{i=1}^n D[i, l(i)]^2 \quad (12)$$

di mana $l(i)$ merupakan klaster yang beranggotakan kasus ke- i .

4. Pemeriksaan, apakah terdapat kasus yang berpindah dari satu klaster ke klaster lainnya yang menghasilkan pengurangan pada E . Pada setiap kasus dilakukan perhitungan menggunakan persamaan (13).

$$R_{l(i), l} = \frac{n(l)D(i, l)^2}{n(l)+1} - \frac{n(l(i))D(i, l(i))^2}{n(l(i))-1} \quad (13)$$

di mana $n(l)$ = jumlah kasus pada klaster ke- l , dan

$l(i)$ = klaster yang beranggotakan kasus ke- i .

5. Berdasarkan perhitungan pada langkah 4, akan diperoleh semua $R_{l(i), l}$ masing-masing kasus positif sehingga tidak ada pergeseran yang menyebabkan nilai di bawah *error*.

Secara umum, prosedur non hirarki dimulai dengan memilih sejumlah nilai kelompok awal sesuai dengan jumlah yang diinginkan kemudian obyek digabungkan ke dalam kelompok-kelompok tersebut. Jika kita sudah memiliki asumsi berapa klaster yang akan kita dapatkan dan kita ingin mengetahui karakteristik klaster tersebut, yaitu observasi mana saja yang termasuk ke dalam kelompok tersebut dan atau ciri dari klaster tersebut, kita bisa menggunakan metoda K-Means, yakni memproses semua objek (kasus) secara sekaligus.

Penggunaan metoda K-Means sangat sensitif terhadap keberadaan *outlier*. Maka dari itu, *outlier* harus benar-benar dihilangkan untuk mendapatkan hasil yang akurat. Tahapan pengelompokkan analisis klaster dengan metoda K-Means adalah sebagai berikut:

- (a) Tentukan banyaknya klaster, dan *centroid*-nya
- (b) Cari jarak setiap obyek dengan *centroid* klaster yang telah ditentukan
- (c) Kelompokkan obyek ke klaster yang paling dekat
- (d) Cari *centroid* yang baru
- (e) Kembali ke langkah 2 sampai tidak terjadi perubahan anggota klaster.

Untuk mengelompokkan data baru ke klaster dengan metoda K-Means :

1. Tuliskan terlebih dahulu klaster yang sudah terbentuk dan peubah-peubahnya.
2. Setiap peubah dalam klaster akan memiliki *mean* tertentu yang sudah distandarisasi nilainya.

3. Data baru juga distandardisasi nilainya sesuai peubah yang sudah didefinisikan dalam klaster.
4. Lalu dihitung nilai jarak *euclidean* dengan cara untuk setiap data dihitung jaraknya pada setiap klaster, lalu jarak yang terkecil berarti memiliki kesamaan terbesar dibandingkan klaster lainnya yang akan menyimpulkan bahwa data tersebut masuk ke dalam klaster tersebut.
5. Sekarang perhitungan *mean* dari *centroid* klaster akan berubah karena bertambahnya data baru sehingga rata-rata akan berubah.

2.4 Evaluasi Temu Kembali

Kinerja temu kembali mempertimbangkan berapa banyak citra yang relevan dan citra yang tidak relevan yang ditampilkan pada sisi pemakai sehingga bisa dihasilkan nilai efektifitas temu kembali citra tersebut yang ditunjukkan pada Tabel 1. Keterkaitan dari setiap citra pada query menurut banyaknya yang dihitung oleh sistem *CBIR* sebagai nilai nyata “bobot” dinyatakan sebagai W dalam cakupan nilai $[0,1]$. Bobot $W=1$ menunjukkan bahwa terdapat relevansi total sedangkan $W=0$ menunjukkan tidak terdapat relevansi. Kebalikan dari nilai relevansi, $1-W$ menunjukkan tidak relevan dari kesamaan citra pada query.

Tabel 1. Keefektifan temu kembali

	Relevansi Hasil	Penyimpangan Hasil
Hasil Kembali n	A_n	B_n
Hasil yang tidak kembali N-n	C_n	D_n

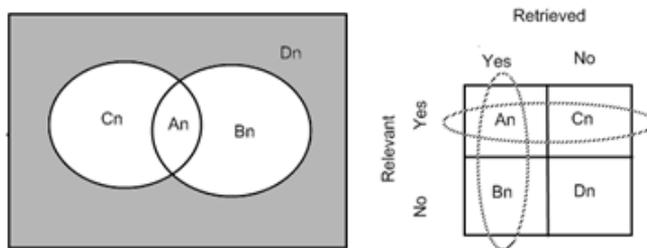
Misalkan N adalah jumlah seluruh citra database dan diurutkan menurun berdasarkan nilai relevan W_r pada query dimana $r = 1,2,\dots,N$ adalah posisi (disebut ranking) dari citra dalam database $W_1 \geq W_2 \geq \dots \geq W_{N-1} \geq W_N$, citra dengan ranking 1 memiliki relevansi yang tinggi dan citra dengan ranking N memiliki relevansi yang minimum. Karena sistem *CBIR* mengembalikan kepada user nilai penggalan tertentu n , $1 \leq n \leq N$, citra dengan relevansi lebih tinggi, kecukupan nilai n secara khusus dievaluasi dengan empat karakteristik berikut :

1. Seluruhnya relevan $A_n = W_1 + \dots + W_n$ dari nilai kembali n (keputusan “benar positif”, atau pendeteksian benar),
2. Tidak relevan sama sekali $B_n = (1 - W_{n+1}) + \dots + (1 - W_n) = n - A_n$ dari nilai keluaran n (keputusan “salah positif”, atau tanda pendeteksian salah),
3. Seluruhnya relevan $C_n = W_{n+1} + \dots + W_N$ dari nilai bukan kembali N-n (keputusan “salah negatif” atau tidak dideteksi), dan
4. Seluruhnya tidak relevan $D_n = (1 - W_{n+1}) + \dots + (1 - W_N) = N - n - C_n$ dari nilai bukan kembali N-n (keputusan “benar positif”, berbeda sama sekali).

Nilai yang disarankan untuk mengukur keefektifan temu kembali :

1. Recall $R_n = A_n / (A_n + C_n)$ jumlah nilai relatif dari hasil relevansi dalam hasil temu kembali n dengan hasil yang relevan dalam databatase
2. Precision $P_n = A_n / (A_n + B_n) = A_n / n$, bagian yang sesuai n kembali dan
3. Fallout atau *false alarm rate* $F_n = B_n / (B_n + D_n)$, angka relatif dari hasil yang benar-benar berbeda).

Gambar 2 menunjukkan diagram Venn keefektifan temu kembali dan hubungan nilai relevansi dan pembacaan.



Gambar 2 Keefektifan temu kembali

Pengukuran tambahan berfokus justru pada temu kembali *item* yang tidak relevan : *missed results* $M_n = C_n / (A_n + C_n)$, nilai relatif dari hasil yang tidak relevan atau disebut juga *inverse recall* : $M_n = 1 - R_n$.

Kinerja temu kembali dari suatu sistem secara kasar dapat dievaluasi dari nilai rata-rata *precision* ditunjukkan pada persamaan (14) dan *recall* ditunjukkan pada persamaan (15) melalui sebuah tolok ukur query. Rata-rata *precision* adalah ukuran evaluasi yang diperoleh dengan cara menghitung rata-rata tingkat *precision* pada berbagai tingkat *recall* [17].

$$Precision = \frac{\text{jumlah citra relevan yang terambil}}{\text{jumlah seluruh citra yang terambil}} \tag{14}$$

$$Recall = \frac{\text{jumlah citra relevan yang terambil}}{\text{jumlah citra relevan dalam database}} \tag{15}$$

Harmonic Mean/F-Measure/F-Score/F1-Score adalah suatu pengukuran yang menggabungkan nilai *precision* dan *recall*, dirumuskan dalam bentuk persamaan (16).

$$F = \frac{2}{\frac{1}{R} + \frac{1}{P}} = \frac{2RP}{R+P} \tag{16}$$

di mana R = nilai *recall*; P = nilai *precision*

Nilai F berada pada rentang [0.1], nilai F yang semakin tinggi jika nilai *precision* dan nilai *recall*nya tinggi. Semakin tinggi nilai F dari suatu sistem temu kembali, berarti semakin baik pula kinerja sistem temu kembali tersebut.

2.5 Database WANG

Database WANG merupakan database citra yang terdiri dari 1.000 citra yang berasal database foto Corel yang dipilih dan dikelompokkan menjadi 10 kelompok dan setiap kelompok terdiri dari 100 citra. Database WANG merupakan database citra standar yang dipergunakan sebagai database untuk mengukur kemiripan pencarian citra pada penelitian yang terkait dengan query CBIR. Pengukuran citra dari database WANG dihitung dengan mencari nilai estimasi relevansi berdasarkan citra masukan dalam bentuk query citra dengan asumsi pengguna mencari citra pada kelas yang sama, sehingga dapat diketahui bahwa 99 gambar yang tersisa dari kelompok yang sama dianggap sebagai citra yang relevan, sedangkan citra yang berasal dari kelompok lain dianggap sebagai citra yang tidak relevan.

3. Pengklasteran dan Query CBIR

Ekstraksi fitur untuk pengklasteran database citra yang merupakan ekstraksi fitur warna citra didapatkan dengan cara melakukan ekstraksi setiap nilai piksel dari citra f berdasarkan sebuah citra dasar g untuk masing-masing komponen warna R, G dan B yang akan menghasilkan nilai MSE_R , MSE_G dan MSE_B . Untuk citra yang memiliki resolusi yang berbeda dengan citra dasar, maka citra tersebut akan di *resize* terlebih dahulu sehingga resolusinya akan sama dengan resolusi citra dasar. Ukuran citra hasil *resize* yang sudah sama dengan citra dasar yang berukuran $M \times N$ piksel, kemudian dihitung nilai MSE dengan menggunakan persamaan (6) untuk masing-masing komponen warna R, G dan B. Untuk masing-masing nilai MSE yang dihasilkan kemudian dipergunakan untuk mendapatkan nilai $PSNR$ dari setiap komponen warna R, G dan B, kemudian ditentukan nilai $PSNR$ yang terkecil sebagai $PSNR$ minimum dirumuskan pada persamaan (17) dan nilai $PSNR$ terbesar sebagai $PSNR$ maksimum dirumuskan pada persamaan (18).

$$PSNR_{Min} = \min_{x=(R,G,B)} \left\{ 10 * \log_{10} \left[\frac{255^2}{MSE_x} \right] dB \right\} \quad (17)$$

$$PSNR_{Max} = \max_{x=(R,G,B)} \left\{ 10 * \log_{10} \left[\frac{255^2}{MSE_x} \right] dB \right\} \quad (18)$$

Implementasi algoritma K-Means untuk menentukan posisi kluster setiap citra dilakukan dengan cara menghitung terlebih dahulu jarak citra ke kluster terdekatnya dengan menggunakan rumus jarak *Euclidean*. Penentuan rumus jarak yang dipergunakan yaitu jarak *Euclidean* didasarkan pada parameter pengklasteran yang dipergunakan yaitu $PSNR$ minimum dan maksimum yang diasumsikan sebagai sebuah koordinat diukur berdasarkan pusat kluster yang akan terbentuk dan dicari jarak terpendek dari kedua titik tersebut. Untuk mendapatkan jarak antara kedua titik data tersebut maka dapat dihitung dengan akar kuadrat jarak dari jumlah kuadrat perbedaan antara nilai-nilai yang sesuai yang dikenal dengan pengukuran jarak *Euclidean*. Rumus

perhitungan jarak *Euclidean* yang secara umum dituliskan dalam bentuk persamaan seperti dituliskan pada persamaan (19).

$$d_{ij} = \sqrt{\sum_{k=1}^m (x_{ik} - x_{jk})^2} \tag{19}$$

dimana d_{ij} = jarak antara objek i dengan objek k
 x_{ik} = nilai objek i pada peubah ke- k
 x_{jk} = nilai objek j pada peubah ke- k
 m = banyaknya peubah yang diamati

atau dituliskan dalam bentuk implementasinya sebagai persamaan (20).

$$d_{ij} = \sqrt{(PSNR_{min}(x_{ik}) - Cmin_{jk})^2 + (PSNR_{max}(x_{ik}) - Cmax_{jk})^2} \tag{20}$$

dimana D = jarak antara objek i dengan objek k
 $PSNR_{min}(x_{ik})$ = nilai $PSNR_{min}$ objek i pada peubah ke- k
 $PSNR_{max}(x_{ik})$ = nilai $PSNR_{max}$ objek i pada peubah ke- k
 $Cmin_{jk}$ = nilai centroid min objek j pada peubah ke- k
 $Cmax_{jk}$ = nilai centroid max objek j pada peubah ke- k

Perhitungan jarak d_{ij} dilakukan dengan cara menghitung jumlah kuadrat dari selisih $PSNR$ minimum dengan pusat kluster minimum ke- j ditambah dengan selisih $PSNR$ maksimum dengan pusat kluster maksimum ke- j . Hasilnya kemudian diakarkuadratkan dan akan menghasilkan jarak yang dicari. Hasil penentuan posisi citra pada setiap kluster disimpan pada tabel tblkluster.

Ekstraksi fitur yang dipergunakan dalam query CBIR pengujian menggunakan ekstraksi fitur warna citra untuk mendapatkan nilai intensitas (I), yang merupakan ekstraksi setiap nilai piksel dari citra yang dicari (S) untuk masing-masing komponen warna R, B dan G yang dihitung dengan menggunakan rumus pada persamaan (21). Dengan cara yang sama untuk record citra (T) juga dilakukan ekstraksi setiap nilai piksel untuk masing-masing komponen warna menggunakan rumus pada persamaan (22).

$$I_S = (0,2126R + 0,7152G + 0,0722B)^{(nilai\ R,G,B)} \tag{21}$$

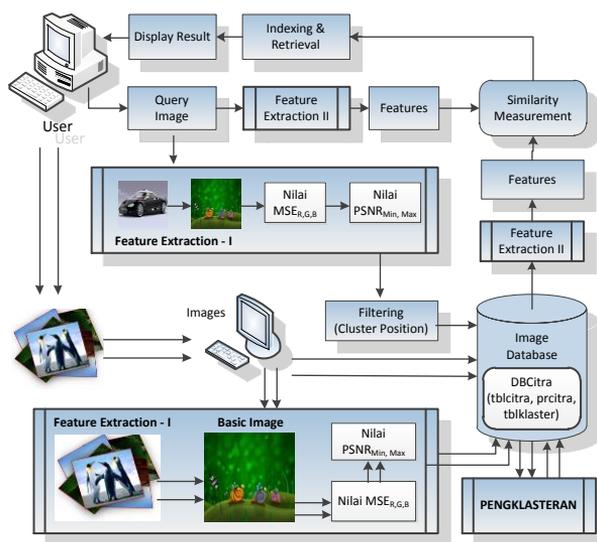
$$I_T = (0,2126R + 0,7152G + 0,0722B)^{(nilai\ R,G,B)} \tag{22}$$

Pada perhitungan kemiripan (*similarity measurement*), perhitungan dilakukan dengan menghitung total harga mutlak selisih nilai intensitas setiap piksel citra yang dicari dengan record citra dibagi dengan perkalian ukuran dimensi citra ($M \times N$) dengan 255 (nilai maksimal setiap komponen warna), hasilnya dinyatakan dalam bentuk prosentase tingkat kemiripan. Perhitungan ini dituliskan dalam bentuk formula jumlah perbedaan mutlak yang dikenal dengan SAD (*sum of absolute differences*) dituliskan pada persamaan (23).

$$SAD = \left[\frac{1}{M \cdot N \cdot 255} \sum_{i=1}^M \sum_{j=1}^N \| I_S(i, j) - I_T(i, j) \| \right] \times 100 \% \tag{23}$$

4. Hasil dan Pembahasan

Gambar 3 menunjukkan arsitektur sistem yang diusulkan, terdapat dua tahapan yang dilakukan untuk implementasi yaitu tahapan penyimpanan file citra sebagai record dalam database citra kemudian melakukan ekstraksi fitur warna setiap record citra untuk mendapatkan nilai PSNR minimum dan maksimum dilanjutkan dengan proses pengklasteran. Tahapan yang lain adalah pencarian citra itu sendiri berdasarkan masukan citra yang dicari dengan cara melakukan ekstraksi fitur warna yang akan mendapatkan nilai intensitas warna sebagai parameter penentuan posisi kluster dilanjutkan dengan membandingkan tingkat kemiripan citra yang dicari dengan record database citra.



Gambar 3 Arsitektur Sistem

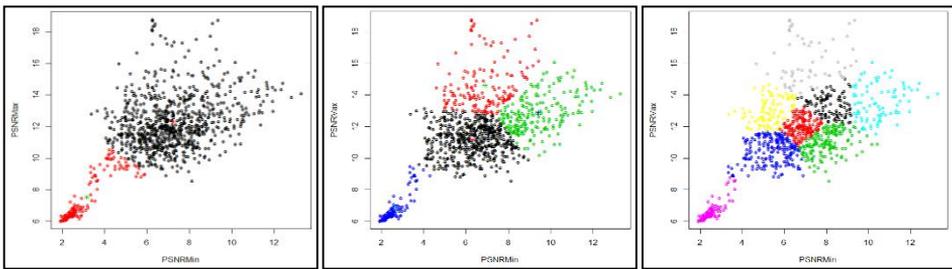
Pusat kluster dan jumlah record pada pembentukan kluster menggunakan algoritma K-Means pada record database WANG yang berjumlah 1.000 record ditunjukkan pada Tabel 2 untuk pembentukan 2, 4 dan 8 kluster yang terdiri kelompok kluster, kluster, nilai *centroid PSNR* minimum dan maksimum, dan jumlah record pada setiap klasternya.

Tabel 2. Hasil Pengklasteran K-Means database WANG

Cluster		Centroid PSNR (db)		Record Count
Group	Number	Minimum	Maximum	
2	1	7,238144	12,298757	838
2	2	3,217335	7,589639	162
4	1	6,318496	11,190640	529
4	2	6,894847	14,625369	155

4	3	2,638533	6,818268	119
4	4	9,449756	12,881852	197
8	1	10,119414	13,778683	110
8	2	8,000599	11,858669	209
8	3	5,688476	11,963560	188
8	4	6,923792	10,427970	147
8	5	7,305264	17,609415	24
8	6	4,749914	9,994102	95
8	7	6,782889	13,985441	121
8	8	2,518710	6,560355	106

Tabel 2 menunjukkan bahwa record yang diproses dalam database WANG sebanyak 1.000 record dengan hasil proses untuk 2 klaster pada klaster ke-1 memiliki nilai *centroid PSNR_{Min}* sebesar 7,238144 db dan *centroid PSNR_{Max}* sebesar 12,298757 db dengan jumlah 838 record, sedangkan klaster ke-2 memiliki nilai *centroid PSNR_{Min}* sebesar 3,217335 db dan *centroid PSNR_{Max}* sebesar 7,589639 db dengan jumlah 162 record. Gambar 4 menunjukkan grafik plot setiap record berdasarkan nilai *PSNR* minimum dan maksimum pada pembentukan 2, 4, dan 8 klaster menggunakan algoritma K-Means database WANG. Setiap klaster di plot dengan warna berbeda.



Gambar 4 Grafik plot pembedakan 2,4 dan 8 klaster database WANG

Evaluasi kinerja query CBIR menggunakan intensitas warna dapat dijelaskan dengan mengambil contoh pencarian citra 412.jpg pada record database WANG sebagai citra S, misalkan yang akan dibandingkan adalah record citra 416.jpg sebagai citra T. Record citra yang dicari yaitu 412.jpg memiliki ukuran 384 x 256 piksel, demikian juga record citra 416.jpg memiliki ukuran 384 x 256 piksel, sehingga antara citra S yang dicari dan record citra T tidak dilakukan *resize* karena keduanya memiliki ukuran yang sama. Hasil ekstraksi fitur warna untuk masing-masing komponen R, G, dan B citra S pada posisi piksel (i, j) sebagai $R_S(i, j), G_S(i, j)$ dan $B_S(i, j)$ dan citra T sebagai $R_T(i, j), G_T(i, j)$ dan $B_T(i, j)$ untuk $i = 1, 2, \dots, 383$, dan $j = 1, 2, \dots, 255$. Untuk setiap posisi piksel (i, j) komponen warna R, G, dan B pada citra S dan T tersebut kemudian dihitung nilai intensitas untuk citra S dan citra T menggunakan persamaan (21) dan (22) kemudian dihitung selisih jarak mutlaknya menghasilkan nilai berikut:

$$I_S - I_T = \sum_{i=1}^{384} \sum_{j=1}^{256} \| I_S(i, j) - I_T(i, j) \| = 2629639,10960043$$

Hasil selisih jarak $I_S - I_T$ kemudian dihitung sebagai nilai SAD yang menghasilkan nilai berikut :

$$SAD = \left[\frac{1}{384 \cdot 256 \cdot 255} \sum_{i=1}^{384} \sum_{j=1}^{256} \| I_S(i, j) - I_T(i, j) \| \right] \times 100 \%$$

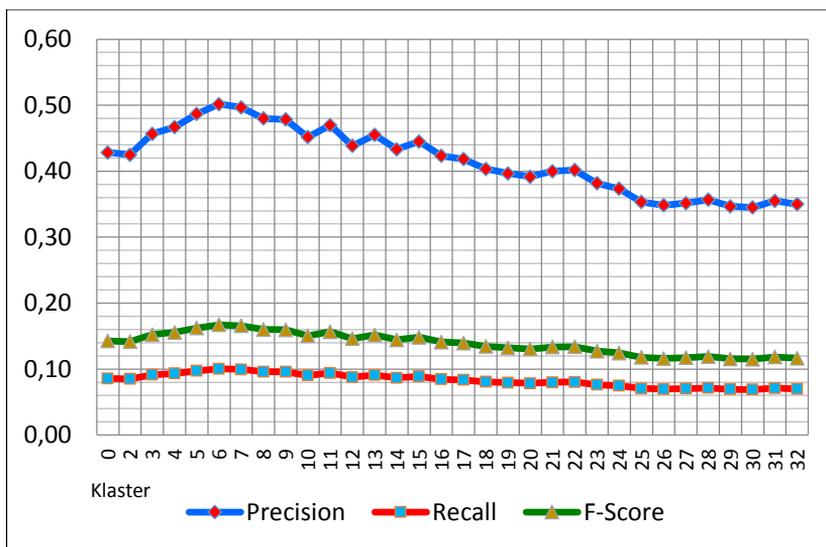
$$SAD = \left[\frac{1}{384 \cdot 256 \cdot 255} \times 2629639,10960043 \right] \times 100 \% = 10,4902244402335$$

Hasil nilai SAD sebesar 10,4902244402335 ini merupakan nilai selisih mutlak intensitas antara citra S sebagai citra yang dicari dan T yang merupakan salah satu record citra database citra. Hasil nilai SAD selanjutnya dipakai sebagai indeks pengurutan tingkat kemiripan dari tingkat kemiripan terendah ke tingkat kemiripan tertinggi.

Tabel 3 Nilai *precision*, *recall* dan *F-score* per kategori query *CBIR* database WANG menggunakan K-Means ekstraksi intensitas warna

Kategori	Non Kluster		3 Kluster		4 Kluster		5 Kluster		6 Kluster		7 Kluster		8 Kluster	
	P	R	P	R	P	R	P	R	P	R	P	R	P	R
Tribal	0,05	0,01	0,08	0,02	0,15	0,03	0,15	0,03	0,23	0,05	0,20	0,04	0,18	0,04
Beach	0,45	0,09	0,57	0,11	0,58	0,12	0,58	0,12	0,50	0,10	0,55	0,11	0,42	0,08
Monument	0,08	0,02	0,13	0,03	0,10	0,02	0,10	0,02	0,13	0,03	0,08	0,02	0,10	0,02
Bus	0,15	0,03	0,25	0,05	0,28	0,06	0,32	0,06	0,47	0,09	0,55	0,11	0,48	0,10
Dinosaur	1,00	0,20	1,00	0,20	1,00	0,20	1,00	0,20	1,00	0,20	1,00	0,20	1,00	0,20
Elephant	0,43	0,09	0,40	0,08	0,37	0,07	0,53	0,11	0,53	0,11	0,52	0,10	0,52	0,10
Rose	0,97	0,19	0,97	0,19	0,97	0,19	0,97	0,19	0,95	0,19	0,95	0,19	0,92	0,18
Horse	0,78	0,16	0,80	0,16	0,88	0,18	0,88	0,18	0,87	0,17	0,88	0,18	0,88	0,18
Mountain	0,30	0,06	0,30	0,06	0,27	0,05	0,27	0,05	0,23	0,05	0,17	0,03	0,23	0,05
Food Dish	0,07	0,01	0,07	0,01	0,07	0,01	0,07	0,01	0,10	0,02	0,07	0,01	0,07	0,01
Rata-rata	0,43	0,09	0,46	0,09	0,47	0,09	0,49	0,10	0,50	0,10	0,50	0,10	0,48	0,10
F-score	0,14		0,15		0,16		0,16		0,17		0,17		0,16	

Hasil dari query *CBIR* pada kategori *dinosaur* dengan nilai *precision* tertinggi 1,00 dan nilai *recall* tertinggi 0,20. Gambar 5 menampilkan hasil perhitungan *precision*, *recall* dan *F-score* per kategori untuk non kluster dan 2, 3, ... , 32 kluster.



Gambar 5. Grafik nilai *precision*, *recall* dan *F-score* per kategori query *CBIR* database WANG menggunakan K-Means ekstraksi intensitas warna

Gambar 5 menunjukkan bahwa query *CBIR* menggunakan 6 dan 7 klaster memiliki kinerja terbaik dengan nilai *F-score* 0,17, sedangkan query *CBIR* yang lainnya memiliki nilai *F-score* kurang dari 0,16.

5. Kesimpulan

Berdasarkan hasil penelitian dan pembahasan yang sudah diuraikan pada bab sebelumnya, maka dapat disimpulkan bahwa :

1. Telah dihasilkan teknik klasifikasi record database citra menggunakan klaster berdasarkan ekstraksi fitur warna record database citra yang dapat dimanfaatkan sebagai pengganti teknik pelabelan.
2. Hasil pengujian implementasi algoritma pengklasteran menggunakan nilai *PSNR* minimum dan maksimum sebagai kunci pengklasteran yang paling memungkinkan untuk dipergunakan adalah algoritma K-Means. Hasil pengklasteran record database citra dalam arsitektur ini dimanfaatkan sebagai indeks klaster pada query *CBIR* menggunakan klaster.
3. Hasil pengujian evaluasi kinerja menunjukkan query *CBIR* menggunakan pengklasteran K-Means proses pencarian citra menggunakan klaster, tingkat akurasi mengalami peningkatan ketika diterapkan pada database WANG yang ditunjukkan dari nilai *F-Score* yang meningkat.

Daftar Pustaka

1. Li, L.J., Wang, G., dan Li, F.F., 2007, *OPTIMOL: automatic online picture collection via incremental model learning*, in Proceedings of the

- 2007 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Minneapolis, MN, USA.
2. Fergus, R., Li, F.F., Perona, P., dan Zisserman, A., 2005, *Learning object categories from Google's image search*, in Proceedings of the 10th IEEE International Conference on Computer Vision (ICCV'05), Beijing, China
 3. Schroff, F., Criminisi, A., dan Zisserman, A., 2007, *Harvesting Image Databases from the Web*, in Proceedings of the 11th IEEE International Conference on Computer Vision (ICCV'07), Rio de Janeiro, Brazil, pp. 2036-2043.
 4. Zitouni, H., Sevil, S., Ozkan, D., dan Duygulu, P., 2008, *Re-ranking of Web Image Search Results using a Graph Algorithm*, in Proceedings of 19th International Conference on Pattern Recognition (ICPR'08), Tampa, FL, USA, Dec. 8-11
 5. Yao T., Mei T., dan Ngo C.W., 2010, *Co-reranking by Mutual Reinforcement for Image Search*, in Proceedings of the ACM International Conference on Image and Video Retrieval (CIVR'10), Xi'an, China, July 5-7, 2010, pp. 34-41.
 6. Jhuo, I.H., dan Lee, D.T., 2010, *Boosting-based Multiple Kernel Learning for Image Re-ranking*, in Proceedings of the 18th International Conference on Multimedia (MM'10), Firenze, Italy, Oct. 25–29, 2010, pp. 1159-1162.
 7. Liu, Y., Mei, T., dan Hua, X.S., 2009, *CrowdReranking: Exploring Multiple Search Engines for Visual Search Reranking*, in Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'09), Boston, Massachusetts, USA, July 19–23, 2009, pp. 500-507.
 8. Jiang, W., Er, G., Dai, Q., dan Gu, J., 2006, *Similarity-Based Online Feature Selection In Content-Based Image Retrieval*. IEEE Trans. Image Processing, 15 (3), pp. 702-712.
 9. Chiu, C., Lin, H., dan Yang, S., 2003, *A Fuzzy Logic CBIR System*. The IEEE International Conference on Fuzzy Systems. pp. 1171- 1176.
 10. Wang, X., dan Xie, K., 2005, *Application of the Fuzzy Logic in Content-based Image Retrieval*. 5(1). pp. 19-24.
 11. Xuelong, H., Li, Y., dan Gensheng, Z., 2007, *Content-based Image Retrieval Using Fuzzy Hamming Distance*. IEEE. The Eighth International Conference on Electronic Measurement and Instruments. 2-826-2-830.
 12. Eckert, M.P., dan Bradley, A.P., 1998, *Perceptual quality metrics applied to still image compression*, Signal Processing 70, pp. 177-200.
 13. Wang, Z., Bovik, A.C., dan Sheikh, H.R., 2004, *Image Quality Assessment: From Error Visibility to Structural Similarity*, IEEE Transactions of Image Processing, vol. 13, pp. 1-12.
 14. Wang, Z. dan Bovik, A.C., 2002, *A Universal Image Quality Index*, IEEE Signal Processing Letters, vol. 9, no. 3, pp. 81-84.

15. Avcibas, I., Sankur, B., dan Sayood, K., 2002, *Statistical evaluation of image quality measures*, Journal of Electronic Imaging, vol 11, no. 2, pp. 206-223.
16. Stephen, K., dan Thompson, J.,D., 2000, *Performance analysis of a new semi orthogonal spline wavelet compression algorithm for tonal medical images*, Med. Phys. 27 : 276-288.
17. Grossman, D.A, Gravano, L., Zhai, C.X, Herzog, O., dan Evans, D.A., 2004, *Statistical precision of information retrieval evaluation*, Proceedings of the 2004 ACM CIKM International Conference on Informations and Knowledge Management, Washington, DC, USA pp. 8-13.

