

SISTEM TEMU KEMBALI CITRA BERBASIS KONTEN MENGGUNAKAN HAAR WAVELET TRANSFORM DAN K-MEANS CLUSTERING

Eriq Muhammad Adams J.¹, Denny Sagita R.²

¹²Fakultas Ilmu Komputer, Universitas Brawijaya
Email: ¹eriq.adams@ub.ac.id, ²denny.sagita@ub.ac.id

(Naskah masuk: 18 Februari 2016, diterima untuk diterbitkan: 17 Maret 2016)

Abstrak

Kombinasi haar wavelet transform, F-Norm, dan progressive retrieval strategy dapat digunakan sebagai metode temu kembali citra berbasis konten yang menghasilkan hasil pencarian yang efektif. Haar wavelet digunakan untuk mendekomposisi citra warna. F-Norm digunakan untuk melakukan ekstraksi fitur. Progressive retrieval strategy digunakan untuk mendapatkan akurasi hasil pencarian yang lebih baik. Pencocokan citra menggunakan progressive retrieval strategy dilakukan terhadap setiap citra yang ada dalam database sehingga menyebabkan waktu pencarian yang lama jika jumlah citra yang ada di database sangat banyak. Dalam penelitian ini diusulkan kombinasi antara haar wavelet transform, F-Norm, progressive retrieval strategy, dan K-Means clustering untuk mempercepat waktu dan nilai precision pencarian. Dari hasil uji coba diperoleh peningkatan nilai precision sebesar 40% serta kecepatan 1,6 – 2,7 kali lebih cepat daripada tanpa menggunakan metode K-Means clustering.

Kata kunci: *Temu Kembali Informasi, CBIR, Haar Wavelet, Fnorm, K-Means, Progressive Retrieval Strategy*

Abstract

The combination haar wavelet transform, F-Norm, and progressive retrieval strategy can be used in content-based image retrieval that produces effective results. Haar wavelet is used to decompose the color image. F-Norm is used to perform feature extraction. Progressive retrieval strategy is used to gain better precision value. Image matching using progressive retrieval strategy is carried out on each image in the database, this causing a long search time if there are lots of images in the database. In this study, we proposed a method based on wavelet transform, F-Norm, progressive retrieval strategy, and K-Means clustering to speed up the search time and increase precision value. From the experiment we obtained precision value increased by 40% and retrieval speed 1.6 to 2.7 times faster than without using the K-Means clustering.

Keywords: *Information Retrieval, CBIR, Haar Wavelet, Fnorm, K-Means, Progressive Retrieval Strategy*

1. PENDAHULUAN

Temu kembali citra berbasis konten merupakan proses temu kembali citra dari *database* yang berisi sekumpulan fitur citra (Yang, 2004). Banyak penelitian dan produk komersial sistem temu kembali informasi citra berbasis konten yang dikembangkan seperti QBIC, Excalibur, dll. (Rui, 1999). Banyaknya topik penelitian mengenai temu kembali citra berbasis konten akibat penyebaran konten video dan citra yang luas dan cepat. Beberapa teknik temu kembali citra berbasis konten selama ini dikembangkan berbasis fitur warna, tekstur, dan bentuk (Nascimento, 2002) (Iqbal, 2002). Fitur warna banyak digunakan dalam teknik temu kembali citra berbasis konten (Kaur, 2016). Sistem temu kembali citra berbasis konten berbasis haar wavelet transform dan progressive retrieval strategy (Latha, 2007) menggunakan fitur warna dalam ekstraksi fitur citra.

Dalam teknik temu kembali citra berbasis konten yang dikembangkan oleh Latha, dkk (Latha, 2007) Haar wavelet transform digunakan untuk mendekomposisi citra ke dalam beberapa level dan koefisien, sedangkan progressive retrieval strategy

digunakan untuk meningkatkan kecepatan dan akurasi dari proses temu kembali citra. Namun, proses temu kembali citra menggunakan progressive retrieval strategy dengan jumlah citra yang sangat banyak membutuhkan waktu yang lama. Oleh karena itu proses temu kembali citra menggunakan progressive retrieval strategy dapat dipercepat menggunakan teknik klasterisasi citra. Algoritma K-Means clustering dapat digunakan sebagai metode klasterisasi citra dalam sistem temu kembali informasi untuk meningkatkan kecepatan dan akurasi dari proses temu kembali citra tersebut (Murthy, 2010).

Berdasarkan penelitian yang telah dilakukan oleh Murthy, dkk (Murthy, 2010). Dalam teknik temu kembali citra berbasis konten yang diusulkan oleh Latha, dkk (Latha, 2007) dapat ditambahkan teknik klasterisasi citra menggunakan algoritma K-Means Clustering untuk meningkatkan akurasi dan kecepatan dalam proses temu kembali citra. Oleh karena itu, dalam penelitian ini diusulkan sistem temu kembali citra berbasis konten menggunakan haar wavelet transform, progressive retrieval strategy, dan K-Means clustering.

2. SISTEM TEMU KEMBALI CITRA BERBASIS KONTEN YANG DIUSULKAN

Sistem temu kembali citra berbasis konten yang diusulkan dalam penelitian ini menggunakan metode klasterisasi citra K-Means *clustering* untuk meningkatkan akurasi dan kecepatan dalam proses temu kembali citra yang dilakukan oleh Latha, dkk sebelumnya (Latha, 2007). Secara garis besar, proses temu kembali citra yang diajukan pada penelitian ini dapat dilihat pada Gambar 1. Pada Gambar 1, proses temu kembali citra dibagi menjadi 2 yaitu proses *online* dan *offline*.

Proses *offline* merupakan proses penyimpanan fitur citra ke dalam *database*. Dalam proses *offline* ini terlebih dahulu citra akan didekomposisi dalam beberapa level dan koefisien menggunakan Haar *wavelet transform*. Citra yang telah didekomposisi dalam beberapa level dan koefisien akan diekstrak menggunakan F-Norm theory menghasilkan fitur citra. Fitur-fitur citra hasil ekstraksi akan diklaster menjadi beberapa kelompok menggunakan algoritma K-Means *clustering*.

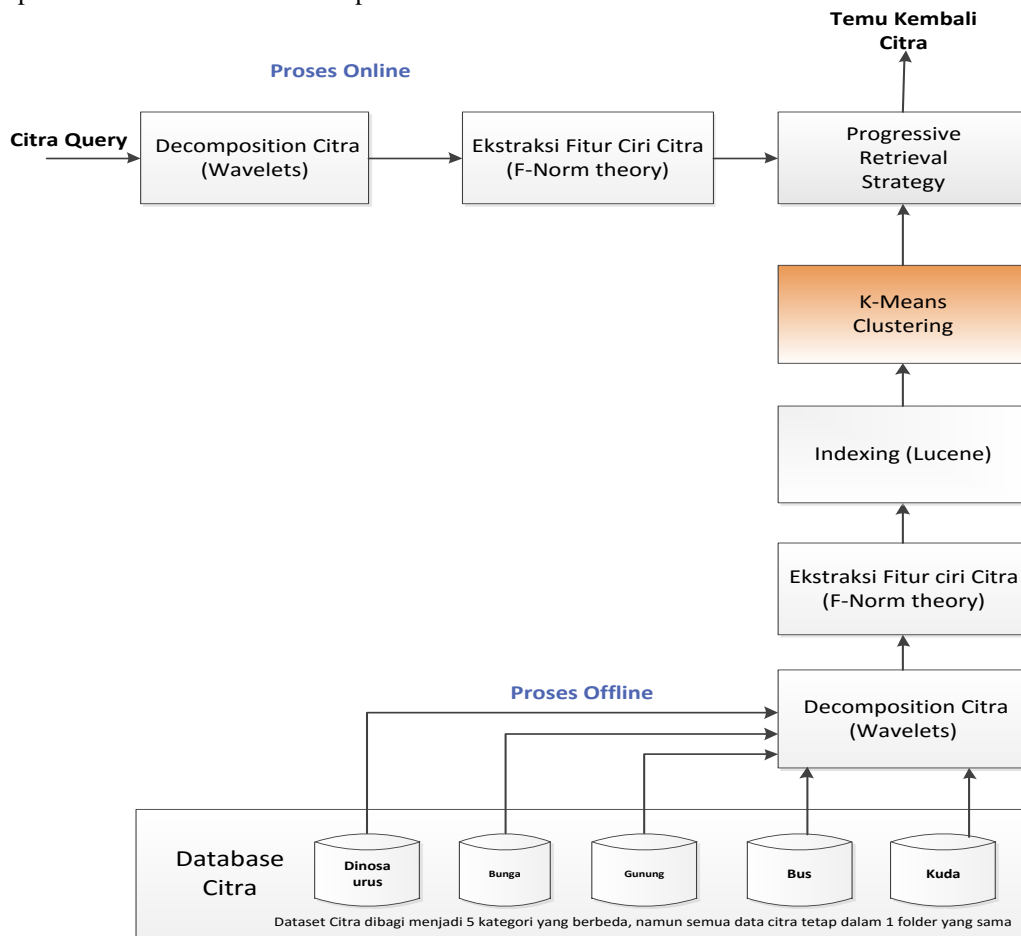
Proses *online* adalah proses pencarian citra yang terdapat dalam database citra. Tahapan dalam

proses *online* ini dimulai dengan dekomposisi citra, ekstraksi fitur, dan pencocokan citra berdasarkan *similarity*-nya menggunakan *progressive retrieval strategy* dalam kelompok atau klaster citra yang memiliki tingkat *similarity* yang paling tinggi.

2.1. Pengambilan Nilai RGB Citra

Sebelum proses dekomposisi citra dilakukan, citra akan diubah dimensinya menjadi 256x256 pixel dalam format RGB untuk mengurangi kompleksitas proses temu kembali citra. Dimensi lebar dan tinggi citra dibuat sama karena proses dekomposisi citra dalam komputasinya membutuhkan *square matrix*. Pengambilan nilai RGB pada citra dapat dilakukan dengan langkah-langkah berikut:

1. Load citra dengan ukuran 256 x 256 pixel.
2. Lakukan proses iterasi sebanyak ukuran luas x tinggi citra untuk mengambil nilai warna pada tiap pixel.
3. Ambil masing-masing nilai red, green, dan blue pada suatu nilai warna RGB di tiap pixel citra dan simpan pada masing-masing array warna yang berkesesuaian.



Gambar 1. Diagram Sistem Temu Kembali Citra yang diajukan

2.2. Dekomposisi Citra dengan Haar Wavelet

Proses dekomposisi citra dengan Haar *wavelet transform* dilakukan sampai n level menghasilkan beberapa koefisien yaitu LL, LH, dan HL (Talukder, 2010) seperti yang ditunjukkan dalam Gambar 2. Proses dekomposisi ini dilakukan terhadap setiap baris dan kolom pada matriks citra dengan ukuran 2^n . Berikut merupakan *pseudocode* dari proses dekomposisi citra menggunakan Haar *wavelet transform*:

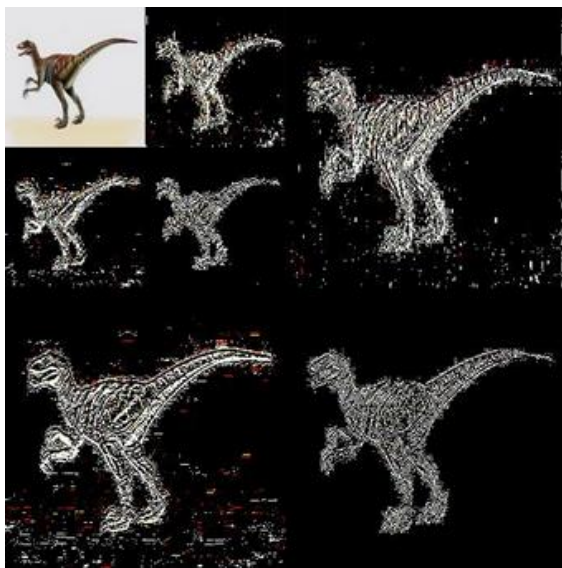
Algoritma Dekomposisi Citra

```

procedure dekomposisi (M, n)
begin
  foreach level n(i) of n
    foreach column c of M
      foreach element c(i) of c
        res(i) = c(i) + c(i+1) / 2
        res(i+1) = c(i) - c(i+1) / 2
        c = res
      endfor
    endfor

    foreach row r of M
      foreach element r(i) of r
        res(i) = r(i) + r(i+1) / 2
        res(i+1) = r(i) - r(i+1) / 2
        r = res
      endfor
    endfor
  endfor
end

```



Gambar 2. Citra yang Didekomposisi Level 2.

2.3. Ekstraksi Fitur

Proses ekstraksi fitur digunakan untuk mengekstraksi fitur citra. Matrik nilai pixel dari suatu citra setelah melewati tahap dekomposisi dalam beberapa level dan koefisien kemudian akan di ekstrak dengan menggunakan F-Norm *theory* (Huang, 2005). Proses ekstraksi fitur ini dilakukan untuk masing-masing koefisien LL, LH, dan HL dalam setiap levelnya. Langkah-langkah dalam proses ini adalah :

1. Inisialisasi matrik dua dimensi hasil dekomposisi pada level n .
2. Menjumlah setiap submatrik dari matrik tersebut. Dimana nilai F-Norm dari matrik tersebut dihitung dengan Persamaan 1.

$$\|A_i\| = \left[\sum_{k=1}^i \sum_{l=1}^i |a_{kl}|^2 \right]^{1/2} \quad (1)$$

A_i merupakan submatrix matrix 2 dimensi, sedangkan k adalah baris dan l adalah kolom.

3. Setelah dilakukan proses F-Norm selanjutnya dihitung ΔA_i yang merupakan elemen dalam vector fitur citra berdasarkan persamaan 2 dan 3,

$$\Delta A_i = \|A_i\|_F - \|A_{i-1}\|_F \quad (2)$$

$$\|A_0\|_F = 0 \quad (3)$$

4. Penyusunan vektor fitur citra dituliskan seperti Persamaan 4, dimana setiap fitur citra .

$$V_{AF} = \{\Delta A_1, \Delta A_2, \dots, \Delta A_n\} \quad (4)$$

2.4. Pengklasteran Citra menggunakan K-Means

Setelah dilakukan proses ekstraksi fitur terhadap semua citra selanjutnya dilakukan proses pengklasteran citra menggunakan K-Means Clustering. Pengukuran jarak dalam K-Means Clustering ini menggunakan persamaan 5. Pengukuran jarak dihitung berdasarkan nilai similarity citra. *Pseudocode* klasterisasi citra menggunakan K-Means dijelaskan dalam *pseudocode* berikut:

Algoritma Klasterisasi Citra

```

procedure kmeans (D)
begin
  while iter < maxIteration
    and distance < epsilon
      centroid = initializeCentroid(D)
      foreach cluster C(i) of C
        foreach image I(i) in cluster C(i)
          selectClosestCluster(I(i),
                                centroid)
        endfor
        distance = calculateMean(centroid)
      endfor
    endwhile
  end

```

$$dist(V_{AF}) = 1 - sim(V_{AF}) \quad (5)$$

2.5. Similarity Citra

Perhatikan Proses ini merupakan penghitungan nilai derajat kesamaan antara sebuah citra pencarian dengan citra-citra yang ada di dalam *image database*. Nilai derajat kesamaan(α) berkisar antara 0 sampai dengan 1. Langkah-langkah prosesnya adalah sebagai berikut :

1. Membaca nilai vektor fitur citra.
2. Load citra *database* yang berisi vektor fitur citra.

3. Bandingkan nilai vektor fitur citra *query* dengan setiap citra pada *image database* untuk mencari nilai *similarity*-nya.
4. Untuk menghitung *similarity* citra *query* dengan citra-citra di dalam *image database* di gunakan persamaan 6 dan 7.

$$\begin{cases} \frac{\min(\Delta A_i, \Delta B_i)}{\max(\Delta A_i, \Delta B_i)} & \text{--- -- } \Delta A_i \neq 0 \text{ atau } \Delta B_i \neq 0 \\ 1 & \text{--- -- } \Delta A_i = 0 \text{ atau } \Delta B_i = 0 \end{cases} \quad (6)$$

$$\alpha = \sum_{i=1}^n C_i \alpha_i \quad (7)$$

5. Hitung nilai koefisien dengan persamaan 8.

$$C_i = \frac{2^{i-1}}{n^2} \quad (i = 1, 2, \dots, n) \quad (8)$$

2.6. Progressive Retrieval Strategy

Dalam *progresive retrieval strategy* (Latha, 2007) dilakukan proses *rough filtering* dan *precise filtering* mulai dari level dekomposisi ke-J sampai level ke-1. Dalam penelitian ini proses pencocokan menggunakan *progressive retrieval strategy* tidak dilakukan terhadap semua *image database* namun hanya terhadap *image database* yang terdapat dalam klaster yang mempunyai jarak antara *centroid* klaster dan citra *query* paling kecil. *Rough filtering* (Hafner, 1995) dilakukan untuk setiap koefisien wavelet LL, LH, dan HL berturut-turut dengan menghitung varian standar (untuk setiap *layer* merah, hijau ,dan biru) dari setiap citra *query* dan *image database* menggunakan persamaan 9.

$$F = (\beta \sigma_r^q < \sigma_r^d < \frac{\sigma_r^q}{\beta}) \&\& (\beta \sigma_g^q < \sigma_g^d < \sigma_g^q / \beta) \&\& (\beta \sigma_b^q < \sigma_b^d < \sigma_b^q / \beta) \quad (9)$$

Dimana β merupakan konstanta filter dengan nilai antara 0 dan 1. Jika nilai F sama dengan FALSE maka *image database* dapat diidentifikasi bahwa *image database* sangat tidak mirip dengan citra *query* sehingga dapat diabaikan. Dalam penelitian ini, nilai varian standar dari *image database* dihitung terlebih dahulu secara offline pada saat indexing untuk mempercepat proses *rough filtering*. *Precise filtering* dilakukan dengan menghitung *similarity* dari fitur koefisien wavelet LL setiap layer merah, hijau, dan biru. Jika hasil *similarity* melewati *threshold* yang ditentukan maka *image database* dapat diabaikan. Setelah proses *progressive retrieval strategy* dilakukan maka dilakukan proses perangkaian hasil pencarian citra berdasarkan *similarity*.

3. HASIL UJI COBA DAN EVALUASI

Pada penelitian sistem temu kembali berbasis konten ini, lingkungan perangkat lunak dan perangkat keras yang digunakan untuk pengembangan aplikasi menggunakan Bahasa Pemrograman Java dan library Apache Lucene,

dengan spesifikasi perangkat keras Intel Quad Core 2.4 Ghz dan RAM 2GB. Karena aplikasi dikembangkan dengan menggunakan bahasa java, maka aplikasi yang dikembangkan diberi nama JCBIR (Java Content Based Image Retrieval). Tampilan aplikasi JCBIR ditunjukkan pada Gambar 3.

Sampel citra yang digunakan sebagai sampel pengujian diperoleh dari situs <http://wang.ist.psu.edu>. Pengujian Citra yang digunakan adalah citra true color 24 bit beresolusi 256 x 256 pixel. Dalam penelitian ini digunakan 500 data citra yang dibagi menjadi 5 kategori, yaitu kategori dinosaurus, bus, pantai, kuda, serta bunga dan setiap kategori terdiri dari 100 citra. Gambar 4 menunjukkan citra bus yang digunakan dan hasil pencarian citra bus. Tiap citra didekomposisi nilai warna pada tiap-tiap pixel dengan membagi menjadi 3(tiga) layer, yakni layer merah, layer hijau, dan layer biru. Sehingga tiap layer warna akan di proses secara terpisah menggunakan Haar *Wavelet Transform* (HWT) untuk didekomposisi dari level ke-1 hingga level ke-4. Hasil dekomposisi berupa nilai koefisien LL, LH, dan HL. Setelah itu hasil dekomposisi yang berupa array dua dimensi akan dinormalisasi nilainya menggunakan teori F-norm untuk menghasilkan vektor nilai fitur-fitur warna dari citra tersebut.

Penelitian ini difokuskan pada akurasi dan kecepatan temu kembali citra. Oleh sebab itu skenario pengujian dibagi menjadi 2, skenario pertama yaitu menguji *recall*, *precision* dan kecepatan temu kembali citra dengan menggunakan metode transformasi Haar *wavelet transform* & *progressive retrieval strategy* dan skenario kedua menguji *recall*, *precision* dan waktu sistem temu kembali citra dengan menggunakan penggabungan Haar *wavelet transform*, *progressive retrieval strategy* dengan k-means clustering. Nilai *precision* yang dimaksud disini adalah proporsi dari suatu set yang diperoleh yang relevan, sedangkan *recall* adalah proporsi dari semua dokumen citra yang relevan di koleksi termasuk dokumen yang diperoleh. Untuk menghitung nilai *recall* dokumen citra dapat menggunakan persamaan 10. Sedangkan untuk menghitung *precision* menggunakan persamaan 11.

$$recall = \frac{\text{jumlah dokumen citra relevan yang ditemukan}}{\text{total jumlah dokumen citra yang relevan}} \quad (10)$$

$$precision = \frac{\text{jumlah dokumen citra relevan yang ditemukan}}{\text{total jumlah dokumen citra yang ditemukan}} \quad (11)$$

3.1. Uji Coba dengan Haar Wavelet Transform

Pada scenario 1 ini, akan di uji pencarian citra dengan menggunakan salah satu sample citra *query* yang mewakili 100 data citra untuk setiap kategorinya. Setiap citra diuji beberapa kali dengan mengubah nilai parameter konstanta *Rough Filtering* (KRF) dan *Precise Filtering Threshold* (PFT). Hal

ini dilakukan untuk memperoleh konfigurasi terbaik dari perubahan parameter sehingga diperoleh hasil pencarian yang akurat dan cepat.

Nilai *precise filtering threshold* yang digunakan sebesar 0.3 dan perubahan konstanta *rough filtering* 0.3, 0.5, 0.7. Pada Tabel 1 menunjukkan hasil pencarian citra dengan menggunakan transformasi Haar *wavelet transform* dan *progressive retrieval strategy*.

Berdasarkan hasil pengujian pada Tabel 1, citra Dinosaurus memiliki nilai *recall* yang cukup baik, dimana dari hasil ujicoba dengan nilai konstanta *rough filtering* sebesar 0.3 dapat memberikan nilai *recall* sebesar 1 atau bisa dikatakan 100% data citra dinosaurus dapat ditemukan seluruhnya. Namun dari nilai *precision* citra dinyatakan cukup kecil hanya memperoleh nilai 0.25. Untuk nilai *recall* terbaik kedua adalah *recall* dari citra Bus dengan nilai sebesar 0.99 dengan parameter konstanta *rough filtering* sebesar 0.3 dan memperoleh 0.96 dengan konstanta *rough filtering* sebesar 0.5. Sedangkan dengan menggunakan konstanta *rough filtering* sebesar 0.7, hanya memberikan nilai 0.47 karena

berdasarkan ujicoba semakin tinggi nilai konstanta *rough filtering* maka hasil pencarian semakin presisi dan nilai konstanta *rough filtering* citra dalam database yang kurang dari 0.7 maka akan di eliminasi dari hasil pencarian. Pada citra pantai dengan menggunakan konstanta *rough filtering* sebesar 0.3 akan diperoleh nilai *recall* sebesar 0.68, begitu juga dengan nilai *recall* pada perubahan parameter 0.5 dan 0.7 diperoleh hasil yang semakin menurun hal ini dikarenakan obyek citra pantai didominasi oleh intensitas warna biru dan berbagai macam obyek yang memiliki kombinasi warna yang bervariasi. Dari Tabel 1, citra bunga nilai yang rendah pula hal ini dikarenakan citra bunga yang diuji memiliki warna agak gelap dan warna citra bunga itu sendiri yang mendominasi sehingga hasil pencarian bunga kebanyakan hanya ditemukan citra bunga yang mirip dengan warna bunga yang dicari, sedangkan citra bunga yang tersedia dalam database bervariasi. Hal ini yang menyebabkan banyak citra bunga yang relevan dalam kategori bunga tidak ditemukan.

Tabel 1. Hasil Pengujian Temu Kembali Citra Menggunakan Haar Wavelet dan Progressive Retrieval Strategy

No.	Citra	KRF : 0.3 & PFT : 0.3			KRF : 0.5 & PFT : 0.3			KRF : 0.7 & PFT : 0.3		
		Recall	Precision	Speed (s)	Recall	Precision	Speed (s)	Recall	Precision	Speed (s)
1.	Dinosaurus	1.00	0.25	3.80	0.99	0.51	2.58	0.67	0.99	1.89
2.	Bus	0.99	0.23	4.00	0.96	0.38	2.47	0.47	0.94	1.95
3.	Pantai	0.95	0.22	3.97	0.75	0.56	2.31	0.10	0.91	1.60
4.	Kuda	0.76	0.17	4.13	0.44	0.21	2.29	0.10	0.48	1.63
5.	Bunga	0.81	0.18	4.26	0.81	0.40	2.80	0.17	0.50	1.72

Tabel 2. Hasil Pengujian Temu Kembali Citra Menggunakan Haar Wavelet, Progressive Retrieval Strategy, dan K-Means (K=5)

No.	Citra	KRF : 0.3 & PFT : 0.3			KRF : 0.5 & PFT : 0.3			KRF : 0.7 & PFT : 0.3		
		Recall	Precision	Speed (s)	Recall	Precision	Speed (s)	Recall	Precision	Speed (s)
1.	Dinosaurus	1.00	0.99	1.48	0.99	0.99	1.50	0.67	0.99	1.31
2.	Bus	0.76	0.80	1.50	0.71	0.89	1.47	0.34	0.94	1.19
3.	Pantai	0.68	0.57	1.86	0.56	0.82	1.40	0.09	1.00	1.05
4.	Kuda	0.66	0.87	1.39	0.32	0.67	1.20	0.04	0.67	0.97
5.	Bunga	0.66	0.94	1.33	0.30	0.91	1.15	0.13	1.00	0.95

3.2. Uji Coba dengan Haar Wavelet dan K-Means

Pada skenario 2 ini, akan di uji pencarian citra dengan menggunakan salah satu sample citra query yang mewakili 100 data citra untuk setiap kategorinya. Perbedaan dengan skenario sebelumnya adalah terletak pada metode klustering yang ditambahkan sehingga setelah dilakukan proses indexing, fitur citra di klaster berdasarkan fitur ciri citra yang memiliki kedekatan dengan fitur citra yang lain.

Metode ini diharapkan dapat mempercepat proses pencarian karena pencarian hanya dilakukan pada klaster tertentu saja. Dalam ujicoba ini 500 citra yang tersedia dalam database dibagi menjadi 5 klaster, hal ini dikarenakan jumlah kategori dalam database citra terdiri dari 5 kategori citra. Diharapkan dengan menggunakan metode K-Means *clustering* ini, fitur ciri citra yang telah di *indexing* dapat masuk kedalam klaster kategori yang sesuai. Sedangkan maksimum iterasi yang ditetapkan pada pengujian ini sebesar 200.



Gambar 3. Tampilan Aplikasi JCBIR

Berdasarkan Tabel 2 mengenai Hasil Pengujian temu kembali Citra Menggunakan penggabungan Haar Wavelet, Progressive Retrieval Strategy, dan K-Means dapat diketahui perubahan nilai precision, dan speed yang cukup signifikan. Hal terlihat pada citra kuda yang mengalami kenaikan precision mencapai 51% dari metode pada skenario 1 dengan konstanta rough filtering sebesar 0.5. Citra dinosaurus dengan menggunakan konstanta rough filtering sebesar 0.3 mengalami peningkatan precision hingga 74% begitu juga dengan citra pantai mengalami peningkatan precision hingga 70%. Dari segi kecepatan dengan menggunakan metode pada skenario 2, kecepatan pencarian lebih cepat 1,6-2,7 kali dari metode pada skenario 1. Namun dari segi recall rata-rata menurun hingga 15%.

3.3. Evaluasi

Berdasarkan grafik pada Gambar 5 dapat diketahui sistem temu kembali citra yang menggunakan Haar Wavelet mempunyai recall yang lebih baik daripada sistem temu kembali citra yang menggunakan penggabungan Haar Wavelet, Progressive Retrieval Strategy, dan K-Means. Penurunan recall dengan menggunakan metode pada skenario 2 mengalami penurunan hingga 15%. Namun sistem temu kembali citra menggunakan penggabungan Haar Wavelet, Progressive Retrieval Strategy, dan K-Means mempunyai nilai rata-rata

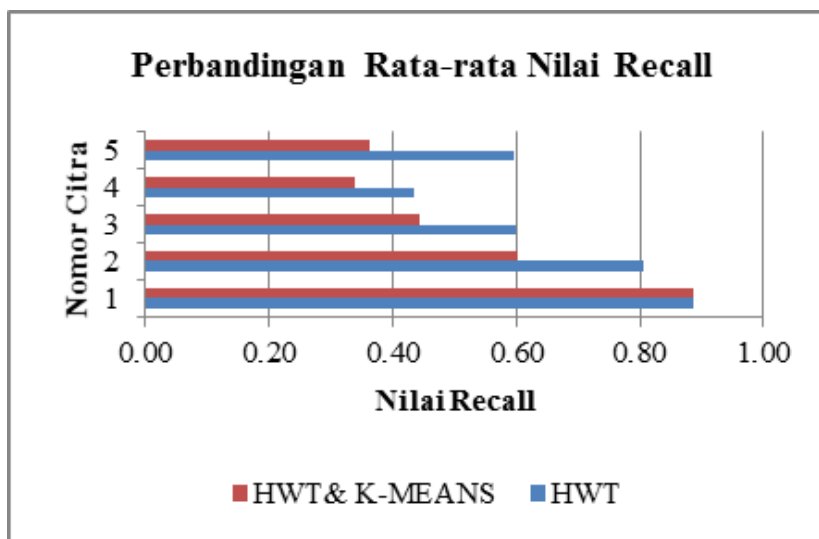


Gambar 4. Hasil Pencarian citra Bus

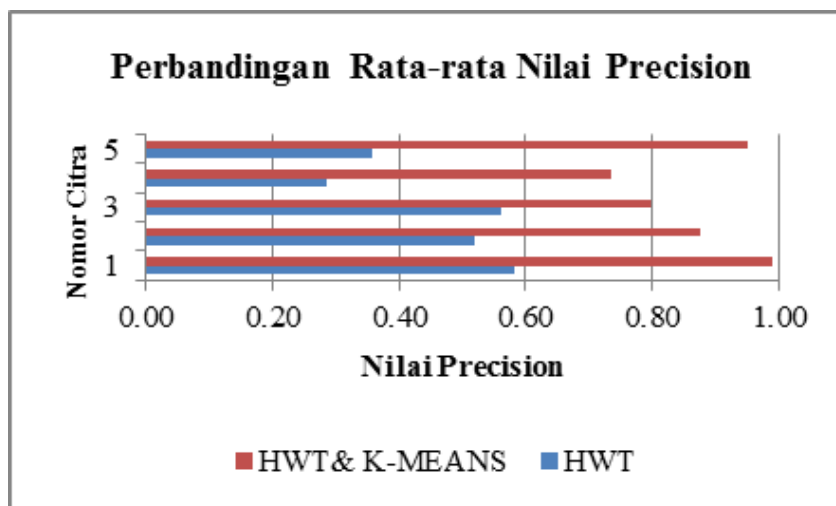
precision yang lebih yang terlihat pada grafik Gambar 6. Peningkatan nilai precision dengan menggunakan metode pada skenario 2 mencapai 40% daripada menggunakan metode pada skenario 1.

Nilai presisi sangat dipengaruhi beberapa faktor, diantaranya adalah parameter konstanta Rough Filtering (KRF) dan Precise Filtering Threshold (PFT). Semakin besar konstanta Rough Filtering menyebabkan eliminasi citra dalam proses pencarian semakin banyak.

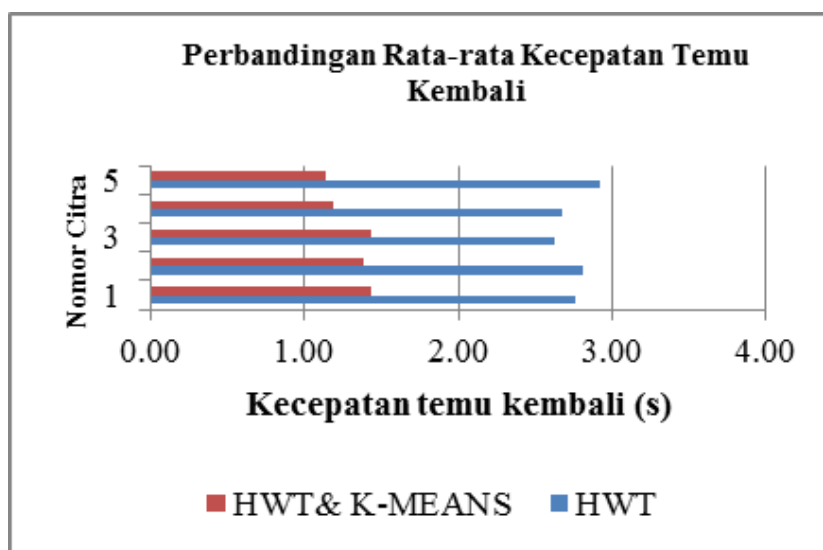
Pada grafik Gambar 7 menunjukkan bahwa dengan menggunakan metode penggabungan Haar Wavelet Transform, *progressive retrieval strategy*, dan K-Means, proses pencarian citra menjadi lebih cepat hingga 1,6-2,7 kali. Hal ini membuktikan hipotesa peneliti benar, bahwa dengan menggunakan metode klustering dapat mempercepat proses pencarian citra menjadi lebih cepat. Tapi ada beberapa hal yang mempengaruhi kecepatan pada proses pencarian, antara lain adalah apabila nilai konstanta Rough Filtering semakin besar maka pencarian akan semakin cepat, namun nilai recall akan semakin menurun. Selain itu faktor perangkat keras yang digunakan, juga mempengaruhi kinerja proses pencarian dan proses indexing citra karena setiap spesifikasi perangkat keras yang berbeda memiliki kecepatan komputasi data yang berbeda pula.



Gambar 5. Perbandingan Rata-rata Nilai Recall antara Skenario 1 & Skenario 2



Gambar 6. Perbandingan Rata-rata Nilai Precision antara Skenario 1 & Skenario 2



Gambar 7. Perbandingan Rata-Rata Kecepatan pencarian antara Skenario 1 & Skenario 2

4. KESIMPULAN DAN SARAN

Dari hasil analisis, perancangan, implemmentasi dan pengujian sistem temu kembali citra dengan menggunakan kombinasi transformasi Haar wavelet transform dan K-means *Clustering* dapat disimpulkan :

1. Sistem temu kembali citra berbasis konten yang diusulkan dengan menambahkan klusterisasi citra menggunakan algoritma K-Means *Clustering* yang dikombinasikan dengan Haar Wavelet Transform dan *progressive retrieval strategy* dapat menambah kecepatan proses temu kembali citra sampai 2,7 kali lipat dan tingkat akurasi (*precision*) 1,4 kali lipat lebih akurat dari pada metode temu kembali citra sebelumnya (Latha, 2007).
2. Dalam proses klusterisasi citra menggunakan K-Means digunakan perhitungan jarak berbasis nilai *similarity* citra yang dihitung menggunakan F-Norm *Theory*.

Untuk meningkatkan akurasi hasil pencarian, dalam penelitian lebih lanjut dapat digunakan algoritma transformasi wavelet lainnya dan algoritma klustering lainnya.

5. DAFTAR PUSTAKA

HAFNER, J., SAWHNEY, H. S., EQUITZ, W., FLICKNER, M., & NIBLACK, W. 1995. Efficient color histogram indexing for quadratic form distance functions. *IEEE transactions on pattern analysis and machine intelligence*. 17(7). 729-736.

HUANG, H., HUANG, W., LIU, Z., CHEN, W. AND QIAN, Q.. 2005. Content-based color image retrieval via lifting scheme. *In Proceedings Autonomous Decentralized Systems*, IEEE. ISADS 2005. 378-383.

IQBAL, Q., & AGGARWAL, J. K. 2002. Combining structure, color and texture for image retrieval: A performance evaluation. *In Pattern Recognition, 2002. Proceedings. 16th International Conference on*. IEEE Vol 2. 438-443.

KAUR, H. 2016. A Review on Comparison of Techniques of Content Based Image Retrieval. *International Journal of Research*, 3(10). 1302-1307.

LATHA, M.Y., JINAGA, B.C. AND REDDY, V.S.K. 2007. Content Based Color Image Retrieval via Wavelet Transforms. *IJCSNS International Journal of Computer Science and Network Security*. 7(12). 38-44.

MURTHY, V.S.V.S., VAMSIDHAR, E., KUMAR, J.S. AND RAO, P.S. 2010. Content Based Image Retrieval using Hierarchical and K-Means Clustering Techniques. *International Journal of Engineering Science and Technology*. 2(3). 209-212.

NASCIMENTO, M. A., & CHITKARA, V. 2002. Color-based image retrieval using binary signatures. *In Proceedings of the 2002 ACM symposium on Applied computing*. ACM 2002. 687-692.

RUI, Y., HUANG, T. S., & CHANG, S. F. 1999. Image retrieval: Current techniques, promising directions, and open issues. *Journal of visual communication and image representation*. 10(1). 39-62.

TALUKDER, K.H. AND HARADA, K.. 2010. Haar wavelet based approach for image compression and quality assessment of compressed image. arXiv preprint. arXiv:1010.4084.

YANG, C. C. 2004. Content-based image retrieval: a comparison between query by example and image browsing map approaches. *Journal of Information Science*. 30(3). 254-267.