

## EKSTRAKSI FITUR BERBASIS WAVELET PADA SISTEM TEMU KEMBALI CITRA TEKSTUR

Nanik Suciati

Jurusan Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember

Email: nanik@if.its.ac.id

**Abstrak** – Pada penelitian ini diimplementasikan suatu sistem temu kembali citra tekstur dengan ekstraksi fitur berbasis wavelet. Ekstraksi fitur terdiri dari 2 proses utama, yaitu dekomposisi wavelet terhadap data citra, dan perhitungan energi dan deviasi standar terhadap koefisien-koefisien wavelet hasil dekomposisi. Penelitian ini bertujuan untuk mengetahui performance sistem temu kembali citra tekstur dengan menggunakan beberapa vektor fitur yang diekstrak dari beberapa level dekomposisi wavelet yang berbeda. Untuk uji coba, dibuat database citra menggunakan citra tekstur Brodatz. Filter yang digunakan pada proses dekomposisi diturunkan dari fungsi wavelet Daubechies4, dan untuk perhitungan kemiripan antara citra contoh dan citra dalam database digunakan Canberra distance. Hasil ujicoba menunjukkan bahwa performance terbaik dari sistem temu kembali citra tekstur adalah recall sebesar 92,5%, yang didapat dengan menggunakan vektor fitur yang diekstrak dari hasil dekomposisi wavelet sampai dengan level 2.

**Kata kunci:** Temu kembali citra tekstur, *Ttransformasi wavelet*, *Canberra distance*

### 1. PENDAHULUAN

Kemajuan teknologi pencitraan digital dan murah nya harga media penyimpan menyebabkan jumlah citra digital dalam *database* meningkat tajam [7]. Beberapa di antara *database* tersebut berisi citra dengan jumlah jutaan. Melakukan *browsing* pada *database* citra berukuran besar untuk menemukan citra-citra tertentu adalah suatu pekerjaan yang sangat tidak efisien. Diperlukan suatu alat otomatis untuk mencari citra pada *database* berdasarkan isi semantik citra. Teknik-teknik awal untuk menemukan citra pada *database* didasarkan pada pencarian anotasi teks yang dibubuhkan pada citra, bukan pada isi (*content*) citra. Salah satu kendala pada penemuan citra berdasarkan anotasi teks adalah kesulitan dalam penyusunan anotasi teks yang dapat mendeskripsikan suatu citra secara tepat dan lengkap. Citra yang sama bisa diinterpretasikan dengan cara berbeda sehingga menghasilkan anotasi teks yang berbeda pula. Pembubuhan anotasi teks pada citra merupakan pekerjaan yang menyita waktu, dan bersifat subyektif, kadang-kadang tidak akurat dan tidak lengkap. Oleh karena itu, temu kembali citra berbasis teks selain

tidak efisien juga tidak praktis untuk *database* citra berukuran besar.

Untuk mengatasi kekurangan dari sistem temu kembali citra berbasis teks, dibutuhkan suatu teknik pengindeksan citra secara otomatis berdasarkan isi visual dari citra, seperti warna, tekstur, dan bentuk. Sistem yang baru ini dikenal dengan sistem temu kembali citra berbasis isi (*content-based image retrieval/CBIR*), yang mengorganisasi *database* citra berdasarkan isi visual dari citra yang diekstrak secara otomatis, dan yang memungkinkan *query* dilakukan dengan menggunakan citra contoh [2,3,6,8].

Tantangan utama dalam sistem temu kembali berbasis isi adalah menemukan fitur-fitur yang dapat mewakili karakteristik unik dari citra, sehingga fitur tersebut dapat digunakan untuk mengidentifikasi citra secara akurat. Fitur visual yang bisa diekstrak dari suatu data citra adalah tekstur, warna, dan bentuk [6,8]. Pada kebanyakan data citra seperti citra kedokteran, citra natural, citra batik, dan lain-lain, fitur tekstur merupakan fitur yang penting karena citra-citra tersebut bisa dipandang sebagai komposisi dari beberapa tekstur yang berbeda.

Fitur tekstur umumnya diekstrak melalui pendekatan multiresolusi. Beberapa hasil penelitian yang terkait dengan temu kembali

citra, fitur tekstur, dan transformasi wavelet, diuraikan sebagai berikut: Liu dan Picard [4] menggunakan fitur Wold untuk pemodelan dan temu kembali citra; Manjunath dan Ma [5] menggunakan fitur yang dihitung dari koefisien wavelet Gabor untuk mengindeks citra fotografi dan citra satelit; Pada proyek SaFe, Smith dan Chang [9] menggunakan fitur yang diekstrak melalui tranformasi wavelet diskrit untuk temu kembali citra; Do dan Vatterli [1] menyusun suatu sistem temu kembali citra berbasis wavelet dengan menggunakan fungsi Gaussian yang digeneralisasi dan pengukuran jarak Kullback-Leibler.

Pada penelitian ini diimplementasikan sistem temu kembali citra tekstur dengan menggunakan transformasi wavelet diskrit. Fitur tekstur dari citra diekstrak dengan menghitung energi dan standard deviasi dari koefisien wavelet. Selanjutnya, kemiripan antara dua citra dihitung dengan menggunakan perhitungan jarak Canberra. Fitur tekstur diekstrak pada level dekomposisi yang berbeda-beda. Tujuannya adalah untuk mengetahui level dekomposisi yang menghasilkan *performance* terbaik, dan untuk mengetahui hubungan antara *performance* sistem temu kembali citra dengan level dekomposisi wavelet.

**2. TRANSFORMASI WAVELET DISKRIT**

Wavelet merupakan suatu metode yang bisa digunakan untuk mendefinisikan ruang multiresolusi. Dengan menggunakan wavelet, suatu ruang vektor bisa didekomposisi ke dalam sekumpulan ruang vektor bersarang dengan resolusi yang berbeda-beda, sehingga memungkinkan dilakukannya analisis terhadap fungsi baik pada domain waktu maupun frekuensi pada resolusi yang berbeda. Saat ini, wavelet telah sukses diaplikasikan pada beberapa bidang, seperti analisis signal, analisis tekstur, pengolahan citra, analisis numerik, grafika komputer, dan lain-lain.

Representasi wavelet pada citra adalah dengan menggunakan transformasi wavelet diskrit 2 dimensi (2D). Sehingga, transformasi wavelet diskrit 1D diaplikasikan sepanjang arah horisontal dan vertikal. Untuk semua

$r, s \in Z$ , didefinisikan fungsi penskala dan fungsi wavelet sebagai berikut:

$$\begin{aligned} \phi_{j,r,s}(x,y) &= \phi_{j,r}(x)\phi_{j,s}(y) \\ \psi_{j,r,s}^H(x,y) &= \phi_{j,r}(x)\psi_{j,s}(y) \\ \psi_{j,r,s}^V(x,y) &= \psi_{j,r}(x)\phi_{j,s}(y) \\ \psi_{j,r,s}^D(x,y) &= \psi_{j,r}(x)\psi_{j,s}(y) \end{aligned} \tag{1}$$

Empat himpunan fungsi  $\{\phi_{j,r,s}\}$ ,  $\{\psi_{j,r,s}^H\}$ ,  $\{\psi_{j,r,s}^V\}$ , dan  $\{\psi_{j,r,s}^D\}$  merupakan basis untuk  $V_j \otimes V_j$ ,  $V_j \otimes W_j$ ,  $W_j \otimes V_j$ , dan  $W_j \otimes W_j$ , dengan  $\otimes$  merupakan notasi *direct product*. H, V, dan D menyatakan bagian horizontal, vertikal, dan diagonal. Selanjutnya,  $V_{j+1} \otimes V_{j+1}$  bisa didekomposisi sebagai berikut:

$$V_{j+1} \otimes V_{j+1} = (V_j \otimes V_j) \oplus (V_j \otimes W_j) \oplus (W_j \otimes V_j) \oplus (W_j \otimes W_j) \tag{2}$$

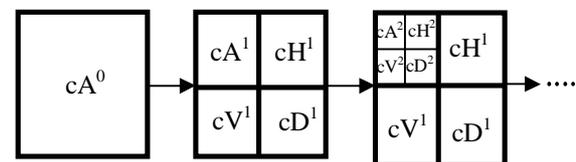
Suatu fungsi dua dimensi  $f_{j+1}(x,y) \in V_{j+1} \otimes V_{j+1}$  bisa didekomposisi menjadi  $f_j(x,y)$ ,  $g_j^H(x,y)$ ,  $g_j^V(x,y)$ , dan  $g_j^D(x,y)$  sedemikian sehingga fungsi semula bisa direkonstruksi kembali dengan rumus berikut:

$$f_{j+1}(x,y) = f_j(x,y) + g_j^H(x,y) + g_j^V(x,y) + g_j^D(x,y) \tag{3}$$

dimana :

$$\begin{aligned} f_j(x,y) &= \sum_{r,s} c_{j,r,s} \phi_{j,r,s}(x,y) \in V_j \otimes V_j \\ g_j^H(x,y) &= \sum_{r,s} d_{j,r,s}^H \psi_{j,r,s}^H(x,y) \in V_j \otimes W_j \\ g_j^V(x,y) &= \sum_{r,s} d_{j,r,s}^V \psi_{j,r,s}^V(x,y) \in W_j \otimes V_j \\ g_j^D(x,y) &= \sum_{r,s} d_{j,r,s}^D \psi_{j,r,s}^D(x,y) \in W_j \otimes W_j \end{aligned} \tag{4}$$

Jika proses dekomposisi yang dihitung dengan rumus-rumus di atas dilakukan secara rekursif, maka akan diperoleh suatu hasil dekomposisi wavelet seperti pada Gambar 1. Nilai  $j = (1,2,\dots)$  merupakan level rekursif dari dekomposisi wavelet.

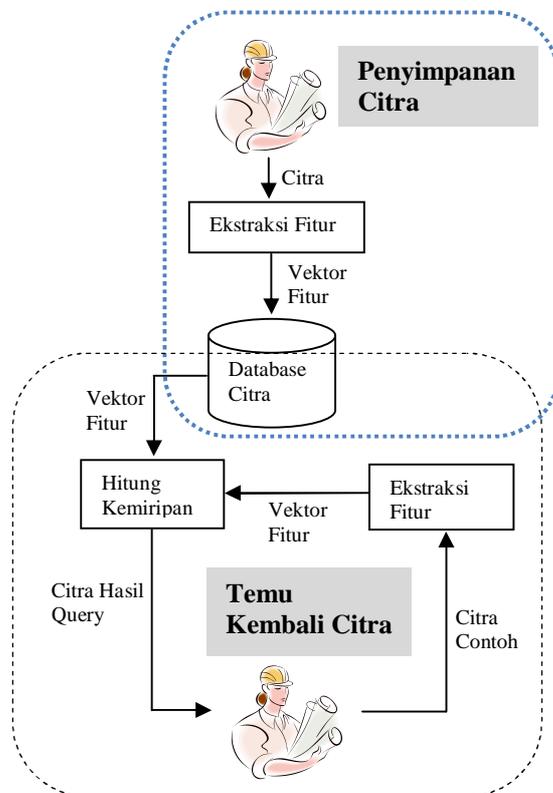


Gambar 1. Proses dekomposisi citra.

**3. SISTEM TEMU KEMBALI CITRA TEKSTUR**

Gambaran umum sistem temu kembali citra tekstur yang diimplementasikan pada penelitian ini ditunjukkan pada Gambar 2. Terdapat dua proses utama yaitu proses

penyimpanan data citra dan proses temu kembali citra. Pada proses penyimpanan citra, data citra tekstur yang di-input-kan oleh *user* disimpan ke dalam *database* bersama dengan vektor fiturnya. Vektor fitur tersebut dihitung di dalam fungsi ekstraksi fitur. Pada proses temu kembali citra, vektor fitur dari citra contoh yang di-input-kan oleh user dihitung kemiripannya dengan vektor fitur citra yang ada di *database*. Citra-citra yang mirip dengan citra contoh akan dikembalikan ke *user*. Dua fungsi penting yang digunakan dalam sistem temu kembali citra adalah fungsi ekstraksi fitur dan fungsi hitung kemiripan. Dua fungsi ini dijelaskan lebih rinci pada subbab berikut.



Gambar 2. Gambaran umum sistem temu kembali citra tekstur.

### 3.1 Fungsi Ekstraksi Fitur

Fitur citra dihitung dari energi dan deviasi standar koefisien-koefisien wavelet. Pertama, dilakukan dekomposisi wavelet terhadap citra tekstur sampai dengan level 5. Pada setiap level dekomposisi, dihasilkan empat buah subcitra, yaitu aproksimasi (cA), informasi frekwensi tinggi arah horisontal (cH), informasi frekwensi tinggi arah vertikal (cV), dan informasi frekwensi tinggi arah diagonal

(cD), yang masing-masing berukuran setengah kali ukuran citra pada level di atasnya (perhatikan Gambar 1). Selanjutnya, dilakukan perhitungan energi dan deviasi standar pada setiap subcitra. Pada setiap level dekomposisi akan didapat empat buah komponen fitur energi dan empat buah komponen fitur deviasi standar. Jika dekomposisi wavelet dilakukan sampai dengan level 1, maka panjang vektor fitur adalah 8. Vektor fitur terpanjang didapat jika ekstraksi fitur dilakukan pada hasil dekomposisi wavelet sampai dengan level 5, yaitu 40.

Energi digunakan sebagai fitur karena distribusi energi pada domain frekwensi dapat mengidentifikasi sebuah tekstur. Selain menghasilkan unjuk kerja yang cukup baik pada *database* tekstur berukuran besar, pendekatan berbasis energi juga didukung oleh studi psikologi tentang *visual cortex*. Dalam penelitiannya, Manjunath [5] dan Kokare [6] menunjukkan bahwa unjuk kerja sistem temu kembali yang menggunakan kombinasi fitur energi dan deviasi standar selalu lebih baik dibandingkan dengan yang hanya menggunakan satu fitur saja. Energi dan deviasi standar dari koefisien-koefisien wavelet dihitung dengan persamaan berikut:

$$E = \frac{1}{M \times N} \sum_{i=1}^M \sum_{j=1}^N |X_{ij}| \tag{5}$$

$$Std = \left[ \frac{1}{M \times N} \sum_{i=1}^M \sum_{j=1}^N (X_{ij} - \mu_{ij})^2 \right]^{\frac{1}{2}} \tag{6}$$

dengan  $M \times N$  adalah ukuran subcitra hasil dekomposisi,  $x_{ij}$  adalah koefisien wavelet, dan  $\mu_{ij}$  adalah nilai rata-rata koefisien wavelet.

### 3.2 Fungsi Hitung Kemiripan

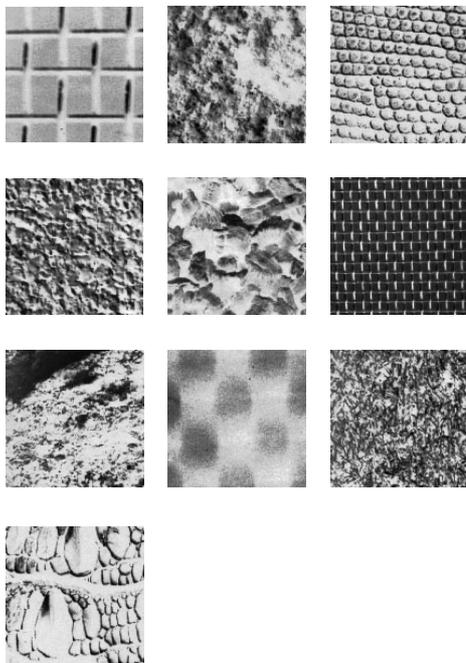
Untuk menemukan kembali data citra yang tersimpan dalam *database*, *user* memberikan input berupa citra contoh. Kemiripan antara citra contoh dengan citra dalam *database* ditentukan berdasarkan perhitungan jarak antara vektor fitur dari citra contoh dan vektor fitur dari citra dalam *database*. Dua buah citra dikatakan mirip jika jarak vektor fitur dari dua citra tersebut relatif dekat. Jika  $x$  dan  $y$  adalah dua buah vektor fitur berdimensi  $d$ , jarak Canberra didefinisikan sebagai berikut:

$$C_{amb}(x, y) = \sum_{i=1}^d \frac{|x_i - y_i|}{|x_i| + |y_i|} \quad (7)$$

Pada persamaan 7, pembilang menyatakan selisih (jarak) dan penyebut menormalisasi selisih tersebut. Dengan demikian, jarak dari setiap komponen fitur tidak akan melebihi satu. Hal ini menjadikan jarak Canberra dapat digunakan untuk menghitung kemiripan antara dua buah citra dengan baik.

#### 4. UJI COBA DAN ANALISIS

Untuk uji coba, dibuat *database* tekstur dengan menggunakan 10 buah citra tekstur yang diambil dari koleksi tekstur Brodatz. Sepuluh buah citra tekstur tersebut ditunjukkan pada Gambar 3. Setiap citra tekstur dengan ukuran asli 640 x 640 piksel dipotong (*crop*) menjadi 16 subcitra yang masing-masing berukuran 128 x 128 piksel. 14 subcitra disimpan di dalam *database*, sedangkan 2 buah subcitra digunakan sebagai citra contoh pada saat melakukan *query*. Total citra dalam *database* adalah 140 buah dan total citra contoh untuk testing adalah 20 buah.



Gambar 3. Citra tekstur yang digunakan untuk uji coba.

Perhitungan vektor fitur dari setiap citra diawali dengan dekomposisi wavelet. Pasangan filter lowpass Lo\_D dan highpass Hi\_D yang digunakan pada proses dekomposisi diturunkan dari fungsi wavelet Daubechies4. Proses dekomposisi dilakukan sampai dengan 5 level. Dekomposisi wavelet level 1 pada citra asli berukuran 128 x 128 menghasilkan 4 buah *subband* cA<sup>1</sup>, cH<sup>1</sup>, cV<sup>1</sup>, dan cD<sup>1</sup> yang masing-masing berukuran 64 x 64. Dekomposisi wavelet level 2 pada *subband* cA<sup>1</sup> berukuran 64 x 64 menghasilkan 4 buah *subband* cA<sup>2</sup>, cH<sup>2</sup>, cV<sup>2</sup>, dan cD<sup>2</sup> yang masing-masing berukuran 32 x 32. Jika proses dekomposisi wavelet diteruskan hingga level 5, maka akan dihasilkan cA<sup>5</sup>, cH<sup>5</sup>, cV<sup>5</sup>, dan cD<sup>5</sup> yang masing-masing berukuran 4 x 4. Energi dan deviasi standar dari koefisien wavelet pada setiap *subband* dihitung dengan menggunakan persamaan 5 dan 6. Karena setiap *subband* menghasilkan dua komponen fitur, maka vektor fitur terpanjang dari sebuah citra terdiri dari 40 komponen fitur, yang didapat dari dekomposisi wavelet sampai dengan level 5. Pada Tabel 1 ditunjukkan panjang vektor fitur dan level dekomposisi wavelet yang digunakan pada uji coba.

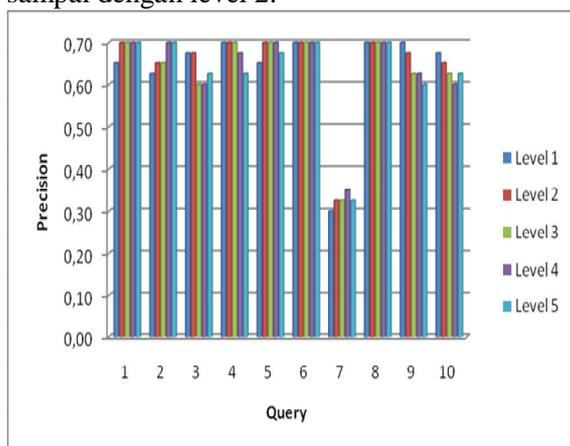
Tabel 1. Level dekomposisi wavelet dan panjang vektor fitur.

Level dekomposisi wavelet	Panjang vektor fitur
1	8
2	16
3	24
4	32
5	40

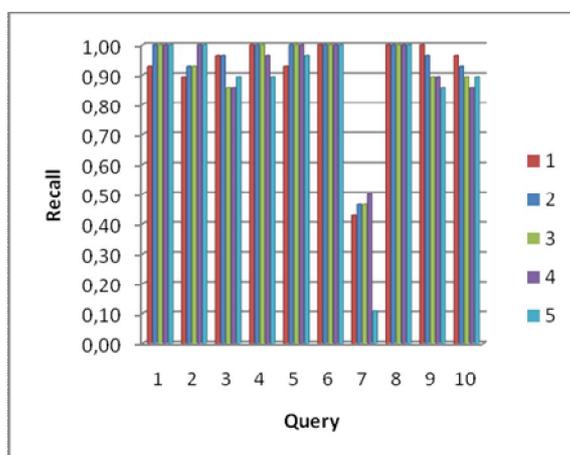
Analisis *performance* dari sistem temu kembali citra tekstur dilakukan dengan menghitung *precision* dan *recall*, dengan menggunakan lima buah himpunan vektor fitur yang berbeda. *Precision* adalah perbandingan antara jumlah citra benar yang berhasil ditemukan dengan total jumlah citra hasil *query*. Sedangkan *recall* adalah perbandingan antara jumlah citra benar yang ada di dalam *database*. Gambar 4 dan 5 menunjukkan grafik *precision* dan *recall* dari 20 *query* dengan 5 himpunan vektor fitur yang berbeda. Pada grafik tersebut, *precision* dan

*recall* dari setiap *query* adalah rata-rata *precision* dan *recall* dari dua *query* untuk satu jenis citra tekstur yang sama. Sehingga, 10 *query* yang tercantum pada grafik mewakili 20 *query*. Pada setiap *query*, akan dicari 20 citra yang paling mirip dengan citra contoh.

Seperti yang ditunjukkan pada Gambar 4, *precision* untuk *query* 1 adalah 0.65, 0.7, 0.7, 0.7, dan 0.7, masing-masing untuk himpunan vektor fitur dari hasil dekomposisi pada level 1, 2, 3, 4, dan 5. Hal ini berarti, pada *query* 1 dan himpunan vektor fitur 1, ditemukan 13 buah citra tekstur yang benar dari 20 citra yang hasil *query*. Sedangkan pada *query* 1 dengan himpunan vektor fitur 2, 3, 4, dan 5, ditemukan 14 buah citra tekstur yang benar dari 20 citra yang hasil *query*. Nilai *precision* rata-rata untuk setiap himpunan vektor fitur ditunjukkan pada Tabel 2. *Precision* rata-rata tertinggi adalah 0.648 yang didapat dengan menggunakan himpunan vektor fitur dengan panjang 16 dari hasil dekomposisi wavelet sampai dengan level 2.



Gambar 4. Grafik *precision* untuk 20 *query* dengan 5 himpunan vektor fitur yang berbeda.



Gambar 5. Grafik *recall* untuk 20 *query*

dengan 5 himpunan vektor fitur yang berbeda.

Tabel 2. *Precision* dan *recall* rata-rata dari 20 *query* dengan 5 himpunan vektor fitur yang berbeda.

Level Dekomposisi	<i>Precision</i> rata-rata	<i>Recall</i> rata-rata
1	0,638	0,911
2	0,648	0,925
3	0,633	0,904
4	0,635	0,907
5	0,628	0,896

Seperti yang ditunjukkan pada Gambar 5, *recall* terendah didapat dari *query* 7, yaitu 0.43, 0.46, 0.46, 0.5, dan 0.11, masing-masing untuk himpunan vektor fitur dari hasil dekomposisi pada level 1, 2, 3, 4, dan 5. Pada *query* 7 dan himpunan vektor fitur 1, ditemukan 5 buah citra tekstur yang benar dari 14 citra benar yang ada di *database*. Sedangkan pada *query* 7 dengan himpunan vektor fitur 2, 3, 4, dan 5, masing-masing ditemukan 6, 6, 7, dan 5, buah citra benar dari 14 citra benar yang ada di *database*. Nilai *recall* rata-rata untuk setiap himpunan vektor fitur ditunjukkan pada Tabel 2. *Recall* rata-rata tertinggi adalah 0.925 yang didapat dengan menggunakan himpunan vektor fitur dengan panjang 16 dari hasil dekomposisi wavelet sampai dengan level 2.

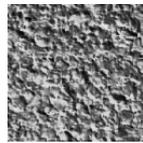
*Precision* dan *recall* menunjukkan bahwa tidak ada kaitan antara panjang vektor fitur dengan performance sistem temu kembali citra. Semakin banyak jumlah fitur yang diekstrak, tidak berarti semakin baik pula performance dari sistem temu kembali citra. Nilai *recall* secara urut dari yang tertinggi adalah 0.925, 0.911, 0.907, 0.904, dan 0.896, masing-masing dari himpunan vektor fitur dengan panjang, 16, 8, 32, 24, dan 40. Secara umum, sistem temu kembali citra tekstur memberikan *performance* yang cukup baik. Dari *query* yang dilakukan, rata-rata ditemukan sebanyak 12 citra benar dari 14 citra benar yang ada di *database*.

Contoh hasil temu kembali citra ditunjukkan pada gambar 6. Citra contoh yang digunakan untuk melakukan *query* ditunjukkan pada Gambar 6(a). Sedangkan 20 citra yang paling mirip dengan citra contoh ditunjukkan pada Gambar 6(b). Dari 20 citra

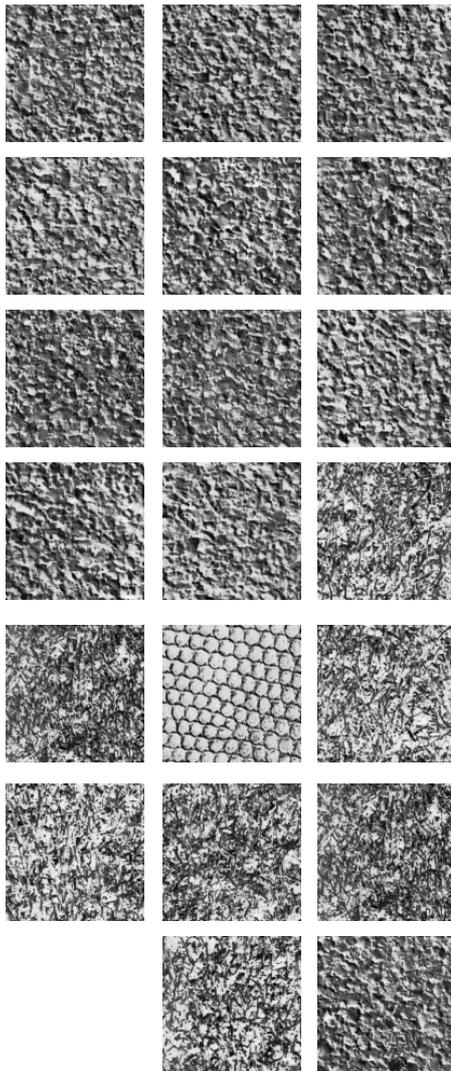
tersebut, 12 citra di antaranya adalah citra yang benar.

## 5. KESIMPULAN

Sebuah sistem temu kembali citra tekstur telah diimplementasikan dalam penelitian ini. Lima himpunan vektor fitur yang berbeda, yang diekstrak dari koefisien wavelet hasil dekomposisi sampai dengan level 1, 2, 3, 4, dan 5, digunakan untuk mengetahui hubungan antara panjang vektor fitur berbasis wavelet dengan *performance* sistem temu kembali citra.



(a)



(b)

Gambar 6. (a) Citra contoh, (b) 20 citra hasil *query*.

Hasil uji coba menunjukkan bahwa tidak ada kaitan antara kedalaman dekomposisi (panjang vektor fitur) dengan *performance* sistem temu kembali citra. *Performance* terbaik didapat dari vektor fitur dengan panjang 16 (hasil dari dekomposisi wavelet sampai dengan level 2), dengan *recall* 92.5%. *Performance* terburuk didapat dari vektor fitur dengan panjang 40 (hasil dari dekomposisi wavelet sampai dengan level 5), dengan *recall* 89.6%. Secara umum, vektor fitur berbasis wavelet cukup dapat mewakili karakteristik unik dari citra tekstur. Kombinasi antara vektor fitur berbasis wavelet dan perhitungan jarak Canberra dapat menghasilkan sistem temu kembali citra tekstur dengan *performance* yang cukup baik.

## PUSTAKA

- [1] Do, M.N., dan Vetterli, M. (2002). Wavelet-based texture retrieval using generalized Gaussian density and Kullback-Leibler distance. *IEEE Transaction on Image Processing*, 11, 146 – 158.
- [2] He, Z., You, X., dan Yuan, Y. (2009). Texture image retrieval based on non-tensor product wavelet filterbanks. *Signal Processing*, 89, 1501–1510.
- [3] Jhanwar, N., Chaudhuri, S., Seetharaman, G., dan Zavidovique, B. (2004). Content based image retrieval using motif cooccurrence matrix. *Image and Vision Computing*, 22, 1211–1220.
- [4] Liu, F., dan Picard, R.W. (1996). Periodicity, directionality, and randomness: Wold features for image modelling and retrieval. *IEEE Trans. Pattern Anal. Machine Intelligence*, 18, 722 - 733.
- [5] Manjunath, B.S., dan Ma, W.Y. (1996). Texture features for browsing and retrieval of image data. *IEEE Trans. Pattern Anal. Machine Intelligence*, 8, 837 – 842.
- [6] Manesh, K., Biswas, P.K., dan Chatterji, B.N. (2007). Texture image retrieval using rotated wavelet filters. *Pattern Recognition Letter*, 28, 1240–1249.

- [7] Shapiro, L.G., dan Stockman, G.C. (2001). *Computer Vision*. Prentice Hall, Upper Saddle River, NJ.
- [8] Shin, Y., Kim, Y., dan Kim, E.Y. (2010). Automatic textile image annotation by predicting emotional concepts from visual features. *Image and Vision Computing*, 28, 526–537.
- [9] Smith, J.R., dan Chang, S.F. (1998). VisualSeek: A fully automated content based image query system. *Proceeding of ACM International Conference Multimedia*, 87 – 98.