

REKONSTRUKSI CITRA KAIN ENDEK BERDASARKAN FITUR TEKSTUR DALAM TEMU KEMBALI INFORMASI CITRA BERBASIS KONTEN

I Gusti Agung Gede Arya Kadyanan

Program Studi Teknik Informatika, Jurusan Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan
Alam, Universitas Udayana
Email : gungde@unud.ac.id

ABSTRAK

Penelitian ini menyajikan rekonstruksi citra kain Endek berdasarkan fitur tekstur. Dalam upaya membangun suatu platform pengembangan sistem perolehan citra warisan budaya seperti citra batik dan lukisan, diperlukan suatu pendekatan yang handal. Keyblock merupakan generalisasi sistem temu kembali berbasis konten pada domain citra. Codebook dibangun berdasarkan pengelompokan nilai keyblock yang mewakili karakteristik citra pada data, kemudian digunakan untuk proses encoding dan decoding. Proses encoding merupakan upaya untuk merepresentasikan setiap sub area pada citra kain Endek dengan suatu nilai indeks yang sesuai dengan codebook. Citra hasil encoding berubah menjadi bentuk matrik 1-dimensi, bentuk ini merupakan analogi dari keyword pada sistem pencarian informasi berbasis teks.

Kata kunci: *keyblock, vektor quantization, codebook, Generalized Lloyd Algorithm, kain Endek*

1. PENDAHULUAN

Fitur tekstur memegang peranan penting pada sistem perolehan (temu kembali), karena tekstur merupakan komponen penentu dari persepsi *visual* manusia dan setiap orang dapat mengenali tekstur walau lebih sulit untuk mendefinisikannya dibandingkan fitur lain. Tidak seperti fitur warna, tekstur lebih menggambarkan wilayah dibandingkan titik atau piksel, dan fitur tekstur ini biasanya didefinisikan hanya berdasarkan tingkat keabuan (*gray level*). Fitur tekstur dapat menggambarkan informasi arah, kekasaran, kontras dan lainnya. Hal tersebut yang membuat fitur tekstur menarik untuk diteliti baik itu terkait dengan cara ekstraksi fiturnya maupun proses pemilihannya.



Gambar 1. Tipikal 6 citra kain Endek yang digunakan dalam penelitian.

Sistem perolehan teks dan perolehan citra sama-sama merupakan suatu sistem perolehan informasi, karena itu teori dan teknik pada perolehan teks yang lebih dulu berkembang

berpotensi juga untuk digunakan pada perolehan citra. Namun generalisasi perolehan informasi dari domain teks ke domain citra tidaklah mudah. Salah satu hambatan besar adalah perbedaan karakteristik intrinsik antara teks dan citra dalam merepresentasikan dan mengekspresikan suatu informasi.

Dalam penyajian informasi secara sintaks, dokumen teks bersifat 1-dimensi, sedangkan citra merupakan bentuk 2-dimensi. Dalam ekspresi informasi secara semantik, satuan informasi dalam bentuk kata dari dokumen teks, terutama kata kunci, memiliki semantik yang terkait dengan dokumen teks. Sebaliknya, satuan informasi dalam bentuk piksel maupun segmen dari citra, tidak mengandung semantik yang secara langsung dengan citra tersebut. Maka pertanyaan besarnya adalah bagaimana membangun segmen-segmen ciri dari citra yang serupa dengan kata kunci pada dokumen.

Studi tentang sistem perolehan citra warisan budaya sebelumnya dilakukan dengan ranah aplikasi citra batik dan studi yang akan datang mungkin akan dilakukan dengan data lukisan. Makalah ini menjelaskan tentang penggunaan *platform* pengembangan sistem perolehan citra berdasarkan *keyblock* untuk citra *kain Endek*. Pengembangan program komputer untuk eksperimen dilakukan dengan menggunakan *software* MATLAB R2009a.

2. TINJAUAN PUSTAKA

2.1. Fitur Tekstur

Tekstur merupakan properti yang dimiliki semua citra. Tekstur berisi informasi penting tentang susunan struktur dari suatu permukaan dan hubungannya terhadap lingkungan sekitarnya. Walau mudah bagi manusia untuk membedakannya, namun sangat sulit bagi komputer digital untuk mendefinisikannya.

Adapun teknik penyimpanan citra digital dalam komputer sebagai array 2-dimensi. Jika $L_x = \{1, 2, \dots, N_x\}$ dan $L_y = \{1, 2, \dots, N_y\}$ adalah data spasial, maka $L_x \times L_y$ adalah himpunan sel-sel resolusi spasial. Citra digital I adalah sebuah fungsi yang memetakan nilai keabuan (*gray tone*) $G \in \{1, 2, \dots, N_g\}$ ke setiap sel resolusi; $I : L_x \times L_y \rightarrow G$. Konsep keabuan berdasar pada variasi kegelapan (*shades*) warna keabuan sel-sel resolusi pada sebuah citra, sedangkan tekstur menekankan pada distribusi spasial (statistik) dari nilai keabuan.

Nilai tekstur dan keabuan memiliki keterikatan yang kuat, dimana keduanya pasti ada pada sebuah citra walaupun salah satunya dapat mendominasi yang lain. Ketika sebuah area kecil dari sebuah citra memiliki sedikit variasi nilai keabuan, maka properti yang dominan adalah keabuan. Namun, ketika variasi nilai diskrit keabuannya meluas, maka properti dominan pada area tersebut adalah tekstur. Hal krusial yang membedakan adalah besaran area, rentang nilai keabuan dan jumlah nilai diskrit keabuan yang dapat dipisahkan. Semakin sedikit jumlah nilai yang dapat dipisahkan, maka keabuan akan dominan. Kenyataannya, ketika area hanya berukuran satu sel resolusi, hanya ada satu nilai diskrit dan properti yang ada hanya keabuan. Ketika jumlah nilai keabuan yang dapat dipisahkan meningkat dalam satu area kecil, maka properti tekstur menjadi dominan [4].

2.2. Pengertian *Keyblock*

Terinspirasi dengan sistem temu kembali informasi berbasis teks, pengusul konsep *keyblock* [5] mencoba untuk menggunakan teori-teori yang dipakai pada temu kembali teks ke dalam domain citra dengan menempatkan *codebook* sebagai *dictionary* dan *keyblock* sebagai *keyword*-nya.

Langkah pertama, berdasarkan ukuran blok yang ditentukan, setiap citra di dekomposisi ulang menjadi sejumlah blok kecil yang berisi fitur dari sub area bersangkutan. Berdasarkan blok-blok tersebut dari contoh citra di basis data, sebuah *codebook* yang berisi *keyblock* dibangun dengan algoritma

pengklusteran VQ (*Vektor Quantizer*) seperti GLA (*Generalized Lloyd Algorithm*) atau PNNA (*Pairwise Nearest Neighbor Algorithm*).

Langkah kedua, setiap citra di basis data diencoding dengan menggunakan *codebook* yang berhasil dibentuk. Dimulai dengan mendekomposisi ulang citra menjadi blok-blok dengan besaran yang sama dengan saat pembentukan *codebook* diatas. Untuk setiap blok, dicari entri terdekat pada *codebook* untuk disimpan indeksnya. Dengan cara tersebut, setiap citra dapat di representasikan sebagai sebuah vektor nilai fitur yang setiap nilainya terhubung dengan indeks *keyblock* pada *codebook*.

Langkah terakhir, semua citra di basis data disimpan dalam bentuk vektor nilai fitur tersebut. Rekonstruksi citra dilakukan dengan mengacu kembali ke *codebook* dan hasil rekonstruksi merupakan pendekatan dari citra aslinya. Dengan menggunakan indeks dari *keyblock*, maka citra menjadi sama dengan dokumen teks yang memiliki struktur 1-dimensi linier. Oleh karenanya, generalisasi sistem temu kembali teks berdasarkan *keyword* dapat digunakan di domain citra berdasarkan *keyblock* [6].

2.3. *Generalized Lloyd Algorithm (GLA)*

GLA adalah algoritma pengklusteran berulang yang memperkirakan kondisi optimal selama perulangan berlangsung saat mendesain vektor kuantisasi. Pada setiap perulangan, langkah pertama, himpunan vektor pelatihan dibagi menjadi beberapa kluster berdasarkan kondisi tetangga terdekat menurut *codebook* yang telah dibentuk saat perulangan sebelumnya, adapun *codebook* awalnya ditentukan secara random. Langkah kedua, *centroid* setiap sel di hitung dan menggantikan vektor *codebook* sehingga *codebook* berisi seluruh *centroid* pada tahap tersebut. Langkah ketiga, keseluruhan *distorsi* dihitung dan perubahan *distorsi* diuji. Jika perubahan *distorsi* melebihi suatu nilai *threshold*, maka proses-proses diatas perlu diulang, jika sebaliknya maka perulangan dihentikan [7]. Adapun langkah-langkah didalam algoritma GLA dapat dijelaskan sebagai berikut [8]:

- Langkah 1 (inisialisasi) :
 - Pilih *training set* $T = \{t_1, \dots, t_l\}$ secara random;
 - Tentukan suatu nilai *threshold* ;
 - Tentukan *codebook* awal, $C_1 = \{c_1, \dots, c_b, \dots, c_N\}$ secara random;
 - Rata-rata *distorsi* awal, $D_0 =$ (suatu nilai yang besar); dan
 - Tentukan jumlah iterasi, $m = 1$.

- Langkah 2 (iterasi pengklusteran)
Untuk $i=1,2, \dots, l$, untuk $j=1, 2, \dots, N$, hitung $d(t_i, c_j)$ dan kemudian berdasarkan ciri tetangga terdekat, hitung

$$P_k = \{t_i \in T \mid k = \arg \min_{1 \leq j \leq N} d(t_i, c_j)\}, \quad (1)$$

untuk mencari klaster baru dari T dengan *distorsi* minimal

$$P = \{P_1, \dots, P_i, \dots, P_N\}$$

- Langkah 3 (perhitungan *distorsi*)
Hitung rata-rata *distorsi* pada himpunan pelatihan terhadap *codebook* saat ini C_m .

$$D_m = \frac{1}{l} \sum_{i=1}^N \sum_{t \in P_i} d(t, c_j) \quad (2)$$

jika $\frac{D_{m-1} - D_m}{D_m} \leq U$, maka iterasi selesai dan

C_m sebagai *codebook* akhir, bila tidak terpenuhi maka iterasi dilanjutkan.

- Langkah 4 (perbarui *codebook*)
Buatlah *codebook* C_{m+1} yang baru dengan menghitung *centroid* dari setiap partisi P_i ;

$$c_i = \frac{1}{|P_i|} \sum_{t \in P_i} t \quad (3)$$

dan ganti *codebook* C_m yang lama dengan *codebook* C_{m+1} , yang baru, perbarui $m = m + 1$ dan lanjutkan ke langkah 2.

2.4. Root Mean Square error (RMS_error)

Untuk melihat kualitas *codebook* yang dihasilkan, maka dilakukan proses *decoding*, yaitu merekonstruksi seluruh citra pelatihan hasil proses *encoding*. Selanjutnya, dihitung *RMS_error* intensitas keabuan antara citra asli dan citra hasil rekonstruksi dengan rumus berikut ini.

$$MSE = \frac{1}{N_1 N_2} \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} (x[n_1, n_2] - \hat{x}[n_1, n_2])^2 \quad (4)$$

Dimana,

N_1 : ukuran baris

N_2 : ukuran kolom

n_1 : koordinat baris

n_2 : koordinat kolom

x : nilai keabuan piksel citra asli

\hat{x} : nilai keabuan piksel citra rekonstruksi

2.5. Pengukuran Distorsi

Dalam penelitian ini, hasil pengukuran *distorsi* juga merupakan suatu faktor pertimbangan penting dalam optimasi pembentukan *codebook*. Dalam proses ini akan dihitung jarak *Euclidean* dari tiap *vektor* pada citra *kain Endek* yang asli dengan *keyblock* yang terkait. Rumus penghitungan *distorsi* dapat dilihat pada Persamaan (2) yaitu pada pembahasan *Generalized Lloyd Algorithm* (GLA). Hasil perhitungan *distorsi* untuk semua ukuran blok dan jumlah *codebook* dapat dilihat pada Tabel 1.

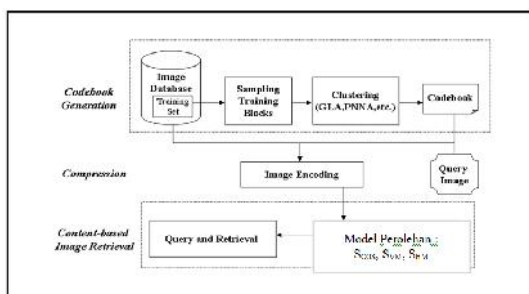
2.6. Model Perolehan Citra Kain Endek

Model yang digunakan dalam perolehan citra *kain Endek* dengan teknik perolehan citra yaitu *quintuple* [D, K, W, Q, S] dimana:

- $D = \{d_1, \dots, d_j, \dots, d_n\}$ adalah daftar citra *kain Endek* terkompresi VQ, dan n disini merupakan jumlah citra *kain Endek* di basis data.
- $K = \{k_1, \dots, k_i, \dots, k_t\}$ adalah daftar *keyblock* pada *codebook*, dan t merupakan jumlah *keyblock* di *codebook*.
- $W : K \times D \rightarrow R^+$ adalah pemetaan yang memetakan pasangan *keyblock* dan citra pada sebuah angka positif. Bila $w_{i,j} = W(k_i, d_j)$, maka W direpresentasikan sebagai matrik $(w_{i,j})_{t \times n}$ disebut bobot matrik yang tiap elemennya adalah indeks *keyblock* pada sebuah citra dimana t adalah jumlah *keyblock* di *codebook* dan n adalah jumlah citra di basis data. Setiap citra d_j , memiliki sejumlah vektor ciri $d_j = \{w_{1,j}, \dots, w_{t,j}\}$.
- $Q = \{q_1, \dots, q_c, \dots, q_l\}$ adalah himpunan citra kueri. Setiap citra kueri memiliki ciri vektor $q = \{w_{1,q}, \dots, w_{t,q}\}$ yang sama dengan vektor ciri d_j .
- $S : Q \times D \rightarrow R^+$ ukuran kesamaan antara kueri dan citra. Nilai ini digunakan untuk membuat rangking citra yang diperoleh.

3. METODOLOGI PENELITIAN

Metodologi yang digunakan dalam pengembangan sistem rekonstruksi citra kain Endek berdasarkan fitur tekstur ini adalah studi literatur dan eksperimen. Studi literatur diambil dari jurnal penelitian dan makalah yang berkaitan dengan pendekatan *keyblock* dan sistem perolehan informasi berbasis konten. Eksperimen dilakukan dengan merealisasikan sistem perolehan informasi berbasis konten dengan pendekatan *keyblock* menggunakan MATLAB R2009a.



Gambar 2. Diagram alir untuk ekstraksi fitur berdasarkan *keyblock* [5].

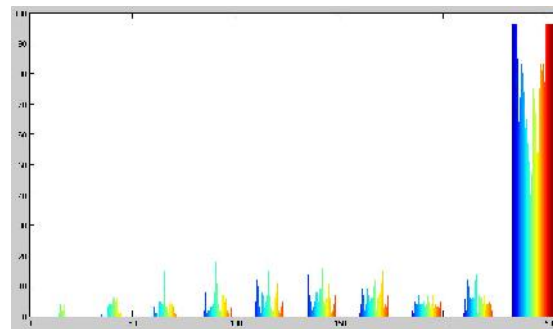
4. HASIL DAN PEMBAHASAN

Dari 328 citra *kain Endek* yang digunakan untuk *training data*, dan sejumlah 41 citra *kain Endek* untuk citra kueri, secara keseluruhan memiliki dimensi citra yang sama yaitu 128 x 96. Dari hasil eksperimen ternyata blok citra dengan ukuran 2x2 yang menghasilkan akurasi yang terbaik dengan beberapa variasi jumlah *codebook* yaitu diantaranya 100, 200, 300, 400, dan 500.

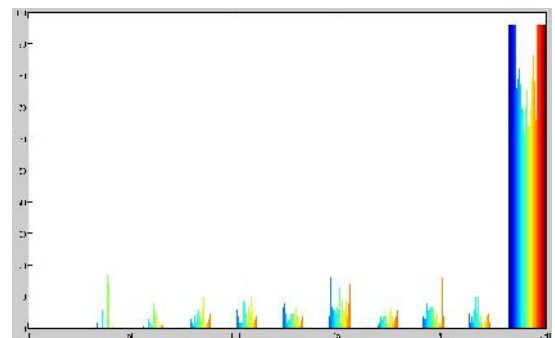
Dari hasil eksperimen ini menunjukkan bahwa *codebook* dapat digunakan sebagai *vector quantifier* untuk proses *encoding* dan *decoding* citra *kain Endek*. Selanjutnya, kualitas citra hasil rekonstruksi diuji dengan rata-rata *RMS_error* terhadap citra aslinya. Dari hasil citra rekonstruksi pada Gambar 5 sampai 7 diperoleh nilai rata-rata *RMS error* yang dihitung dengan Persamaan (4), yaitu masing-masing 9.42, 6.65, dan 6.57 (nilai rata-rata *RMS error* dari seluruh blok dan *codebook* dapat dilihat pada Tabel 2).

Histogram sebuah citra *kain Endek* merupakan representasi grafis dari *frekuensi* intensitas piksel yang ada pada citra tersebut. Pada sebuah citra keabuan 8 bit, nilai keabuan mencapai 256, sedangkan pada citra dengan 16 bit, nilai keabuan mencapai 65.536. Sebuah puncak (*peak*)

pada *histogram* citra menunjukkan kumpulan nilai keabuan yang banyak pada citra, sedangkan lembah (*valley*) menunjukkan kumpulan nilai keabuan yang sedikit pada citra. Gambar 3 dan Gambar 4 berikut menunjukkan tipikal *histogram* untuk citra asli dan citra rekonstruksi dari citra pertama:



Gambar 3. Histogram citra kain Endek asli pertama.



Gambar 4. Histogram rekonstruksi citra pertama.

Kedua *histogram* memiliki 10 *peak* dan setiap *peak* dipisahkan antara 5 – 15 tingkat intensitas keabuan. Dengan jarak antar *peak* yang mencapai 15 tingkat intensitas keabuan, ternyata nilai *RMS_error* yang hanya mencapai 9 tingkat tidak akan mengakibatkan perpindahan *peak* pada tiap piksel yang mengakibatkan perubahan citra yang signifikan. Dengan nilai *RMS_error* yang kecil dan *histogram* yang serupa sehingga antara citra *kain Endek* asli dan citra *kain Endek* rekonstruksi terlihat tidak terlalu berbeda, maka *codebook* yang dibangun dapat digunakan sebagai *vektor quantizer* untuk perolehan citra *kain Endek* berbasis konten.

Seiring dengan bertambahnya ukuran *codebook* menjadi 500 maka nilai rata-rata *RMS_error* turun menjadi 6.57 pada ukuran blok 2x2, yang artinya hasil rekonstruksi masih mirip aslinya. Akan tetapi untuk peningkatan ukuran blok menjadi 4x4 nilai rata-rata *RMS_error* meningkat menjadi 15.83 untuk jumlah *codebook* 100. Peningkatan

ukuran blok ternyata sangat berpengaruh pada kehilangan informasi secara signifikan pada citra. Begitu pula dengan nilai *distorsi*, semakin besar ukuran blok semakin besar nilai *distorsi* yang diperoleh seperti yang dapat dilihat pada Tabel 1.

Tabel 1. Rata-rata *distorsi* pada setiap ukuran *codebook* dan blok

Ukuran blok	Ukuran <i>codebook</i>				
	100	200	300	400	500
2 x 2	30.22	22.23	19.07	21.19	23.02
4 x 4	102.08	92.42	80.85	90.77	84.33
8 x 8	305.34	239.57	239.78	232.73	239.72

Tabel 2. Rata-rata *RMS_error* pada setiap ukuran *codebook* dan blok

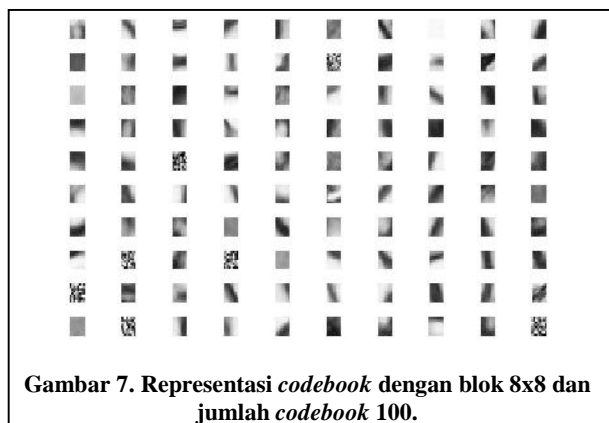
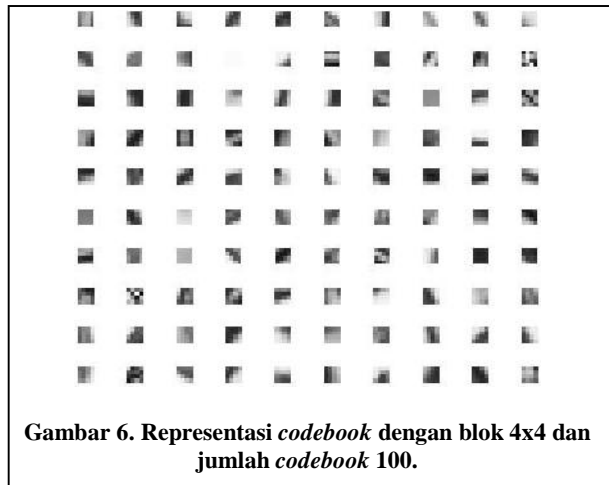
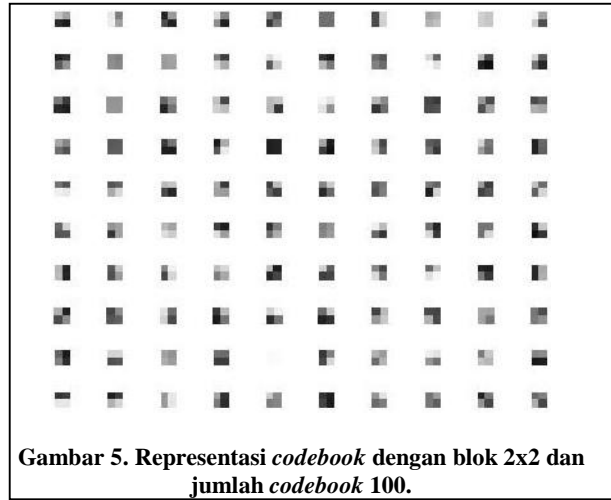
Ukuran blok	Ukuran <i>codebook</i>				
	100	200	300	400	500
2 x 2	9.42	7.52	6.65	6.92	6.57
4 x 4	15.83	14.84	14.20	14.48	13.23
8 x 8	23.58	21.20	20.24	20.23	19.77

Pada Tabel 2 diatas menunjukkan nilai rata-rata *RMS_error* untuk seluruh citra *kain Ende* pada setiap ukuran *codebook* dan *keyblock* yang dibentuk. Dari Tabel 2 terlihat bahwa untuk satu ukuran *codebook*, semakin kecil ukuran *keyblock*, semakin kecil rata-rata *RMS_error* yang terjadi. Untuk satu ukuran *keyblock*, semakin besar *codebook*, semakin kecil *RMS_error* yang terjadi.

Kinerja perolehan cenderung menurun dengan membesarnya ukuran blok. Hal ini sebanding dengan membesarnya rata-rata *distorsi* dan rata-rata *RMS_error* pada ukuran blok yang lebih besar.

Ukuran *codebook* dan ukuran blok untuk efektifitas sistem perolehan citra *kain Ende* dapat dilihat dari nilai rata-rata *distorsi* dan rata-rata *RMS_error* karena semakin kecil nilai *distorsi* dan nilai rata-rata *RMS_error*, maka semakin baik pula kinerja sistem perolehan yang dibangun.

Pada gambar 5 sampai dengan gambar 7 ditampilkan representasi dari *codebook* dengan jumlah 100 dengan ukuran blok 2, 4 dan 8 yang berhasil dibentuk:



5. KESIMPULAN

Hasil studi ini memberikan kesimpulan sebagai berikut:

- 1) Ciri tekstur citra *kain Endek* dapat diekstraksi dengan pendekatan *codebook* dan *keyblok*.
- 2) Berdasarkan keberhasilan didalam sistem temu kembali berbasis text, dengan ini sistem temu kembali informasi berdasarkan konten pada domain citra *kain Endek* dapat dilakukan dengan merepresentasi citra *kain Endek* diubah menjadi vektor ciri 1-dimensi dengan melakukan *encoding* berdasarkan *codebook* yang telah dibentuk.
- 3) Akurasi terbaik diperoleh pada ukuran blok 2x2 dan *codebook* 300, dengan melihat tampilan citra *kain Endek* hasil rekonstruksi secara *visual* dan diperkuat dengan menghitung nilai rata-rata *RMS_error* dapat disimpulkan bahwa kerusakan citra *kain Endek* ketika rekonstruksi tidak terlalu signifikan karena nilai rata-rata *distorsi* 19.07 tingkat keabuan dan nilai rata-rata *RMS_error* yaitu 6.65 tingkat keabuan masih dianggap dalam batas toleransi.

6. DAFTAR PUSTAKA

- [1] J. Eakins dan M. Graham, "Content-based Image Retrieval," *Technology Applications Programme Report 39*, University of Northumbria at Newcastle, October 1999.
- [2] <http://en.wikipedia.org/wiki/CBIR>, Content-based image retrieval" [accessed in October 1, 2009].
- [3] R. M. Haralick, K. Shanmugam, dan I. Dinstein, "Textural Feature for Image Classification," *IEEE Transactions on Sistem, Man dan Cybernetics*, Vol SMC-3 No. 6, November 1973.
- [4] P. Brodatz, "*Textures*", New York, Dover, 1966.
- [5] L. Zhu dan A. Zhang, "*Theory of Keyblock-based Image Retrieval*," *ACM Journal*, pp. 1-32, March 2002.
- [6] L. Zhu, A. Rao, dan A. Zhang, "*Advanced Feature Extraction for Keyblok-based Image Retrieval*," *Proceedings of International Workshop on Multimedia Information Retrieval (MIR 2000)*, Los Angeles, California, USA, November 4, 2000.
- [7] L. Zhu, A. Rao, dan A. Zhang, 2000, "*Keyblok: An approach for content-based geographic image retrieval*," *Proceedings of First International Conference on Geographic Information Science (GIScience2000)*, Savannah, Georgia, USA, 286–287, 2000.
- [8] L. Zhu, C. Tang, dan A. Zhang, "*Using Key blok Statistics to Model Image Retrieval*," *Advances in Multimedia Information Processing – PCM 2001*, Second IEEE Pacific Rim Conference on Multimedia, Beijing, China, October 24-26, 2001, *Proceedings 2001*.
- [9] Yates, R.B, dan B.R. Neto. *Modern Information Retrieval*. Addison Wesley, 1999.