

KOMPRESI CITRA PENGINDERAAN JAUH MULTISPEKTRAL BERBASIS CLUSTERING DAN REDUKSI SPEKTRAL

Agus Zainal Arifin dan Nova Hadi Lestriandoko

Jurusan Teknik Informatika, Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember (ITS) - Surabaya
Kampus ITS, Jl. Raya ITS, Sukolilo – Surabaya 60111
Tel. + 62 31 5939214, Fax + 62 31 5939363
Email : agusza@its-sby.edu

ABSTRAK

Kompresi Citra pada aplikasi Penginderaan Jauh Multispektral merupakan kebutuhan yang sangat vital, sebab citra multispektral merupakan citra yang membutuhkan ruang penyimpanan yang sangat besar. Di sisi lain, citra multispektral memiliki karakteristik istimewa yang dapat dimanfaatkan untuk meningkatkan efektifitas proses kompresinya. Karakteristik ini berkaitan dengan penggunaan citra tersebut dalam proses klasifikasinya. Dengan demikian, kompresi citra multispektral tidak perlu menggunakan cara konvensional, melainkan dengan memanfaatkan karakteristik yang dimilikinya.

Penelitian ini membahas sebuah metode kompresi citra multispektral yang mengintegrasikan metode clustering, manipulasi spektral, serta pengkodean. Metode clustering yang digunakan adalah Improved Split and Merge Clustering (ISM). Pada proses manipulasi spektral digunakan Principal Component Analysis (PCA). Sedangkan untuk pengkodean digunakan metode kompresi data lossless yaitu metode Lempel-Ziv Welch (LZW). Integrasi dari clustering, manipulasi spektral, dan pengkodean ini dibagi menjadi 2 kombinasi, yakni clustering-LZW dan PCA-clustering-LZW.

Evaluasi dilakukan dengan mengukur rasio kompresi, waktu komputasi, jumlah cluster, dan ukuran kesalahan yang meliputi total error, maksimum error, dan rata-rata error. Dari hasil uji coba, didapatkan bahwa masing-masing metode ini memiliki keunggulan yang berbeda pada tiap faktor evaluasinya, sehingga pengguna dapat memilih untuk menggunakan metode yang tepat sesuai kebutuhannya.

Kata kunci : Clustering, kompresi citra, citra multispektral, ISM, reduksi spektral, LZW

1. PENDAHULUAN

Citra multispektral merupakan citra yang terdiri atas sejumlah spektrum. Masalah umum yang muncul dari aplikasi yang berkaitan dengan citra multispektral adalah masalah media penyimpanan serta *bandwidth* yang terlalu besar jika data tersebut akan ditransmisikan ke tempat lain. Apalagi dengan perkembangan teknologi satelit jenis hiperspektral saat ini sudah memungkinkan untuk memperoleh citra hiperspektral dengan 256 kanal, akan menambah kebutuhan ruang penyimpanan yang semakin besar [13]. Teknologi kompresi citra merupakan suatu solusi untuk menyimpan data citra berukuran besar tanpa harus melakukan investasi yang besar untuk meningkatkan ukuran media penyimpanan. Citra yang telah dikompresi mampu berukuran jauh lebih kecil dibandingkan citra aslinya, sehingga akan menghemat waktu dan ruang yang digunakan dalam proses komputer.

Metode kompresi yang telah ada sampai saat ini secara garis besar dibagi menjadi dua jenis yaitu *lossy compression* (ada informasi yang hilang) dan *lossless compression* (tidak ada informasi yang hilang) [14]. Masing-masing jenis kompresi tersebut memiliki kelebihan dan kekurangan. Bila ditinjau

dari rasio kompresi, maka *lossy compression* lebih unggul. Tetapi bila ditinjau dari segi kualitas citra setelah rekonstruksi, maka *lossless compression* lebih baik.

Salah satu metode kompresi citra yang saat ini menjadi standar di dunia internet adalah JPEG [13]. Sampai saat ini standar JPEG masih dikembangkan untuk meminimalkan kesalahan yang timbul pada saat rekonstruksi citra, salah satunya adalah JPEG LS (lossless) [2][13].

Metode JPEG yang saat ini paling banyak digunakan, dapat menghasilkan rasio kompresi yang tinggi dengan mengorbankan struktur detail dari citra. Metode ini sangat cocok diimplementasikan dalam bidang multimedia karena pengamat dapat dengan mudah mengenali citra tanpa memperhatikan struktur detailnya. Tetapi dalam bidang penginderaan jarak jauh, dimana struktur detail dari citra sangat diperhatikan, metode ini tidak sesuai untuk diterapkan [13]. JPEG tidak dirancang untuk melakukan kompresi citra multispektral. JPEG dapat saja digunakan untuk kompresi citra multispektral dengan cara melakukan kompresi citra pada masing-masing kanal [13], yang tentu saja cara tersebut tidak praktis.

Salah satu metode kompresi citra multispektral yang telah dikembangkan oleh Arto Kaarna dan kawan-kawan yaitu metode kompresi citra multispektral menggunakan *clustering* dan reduksi spektral [3]. Algoritma *clustering* yang dipakai adalah *split clustering*. Untuk reduksi spektral digunakan 3 macam metode yaitu PCA (*Principal Component Analysis*), ICA (*Independent Component Analysis*), dan Wavelet. Penggunaan kombinasi metode-metode tersebut pada proses kompresi dapat menghasilkan hasil yang bervariasi.

Salah satu karakteristik citra multispektral adalah adanya korelasi yang tinggi pada tiap bandnya [13]. Karena adanya korelasi yang tinggi tersebut, maka pada sebuah citra multispektral selalu terdapat redundansi informasi. Proses *clustering* dapat digunakan untuk mengurangi redundansi tersebut [3]. Tetapi ternyata dari hasil proses *clustering* masih terdapat redundansi informasi. Pada penelitian yang telah ada [3], proses *clustering* ini dikombinasikan dengan reduksi spektral untuk meminimalkan redundansi informasi.

Metode *clustering* yang digunakan pada penelitian diatas [3] mempunyai beberapa kelemahan, antara lain pembentukan *cluster* berdasarkan pada informasi lokal dan bukan informasi global, maksudnya yaitu pembentukan *cluster* hanya berdasarkan *pixel* yang sudah dimasukkan secara sekuensial, bukan berdasarkan seluruh *pixel* pembentuk citra. Selain itu metode *clustering* ini membutuhkan waktu yang cukup lama.

Oleh karena itu, pada penelitian ini dikembangkan suatu metode kompresi citra yang mengintegrasikan metode *clustering* dan reduksi spektral. Metode *clustering* yang digunakan adalah *Improved Split and Merge Clustering* (ISMC) [5]. ISMC merupakan hasil pengembangan dari algoritma *Split and Merge Classification* (SMC) yang telah berhasil diaplikasikan untuk mengukur suhu permukaan laut (*Sea Surface Temperature*).

Penelitian ini akan berusaha memanfaatkan redundansi informasi pada citra multispektral untuk kompresi. Selanjutnya akan mengintegrasikan algoritma *clustering* dan pengkodean. Tujuannya adalah mencari metode kompresi yang efisien dari segi waktu komputasi, rasio kompresi, dan tingkat kesalahan dengan memanfaatkan metode *clustering* dan reduksi spektral.

2. CITRA MULTISPEKTRAL

Pada bagian ini akan dibahas proses *clustering*, reduksi spektral, dan pengkodean yang akan digunakan pada proses kompresi citra multispektral ini.

2.1. Clustering

Clustering merupakan suatu proses klasifikasi citra penginderaan jauh secara *unsupervised* (tak terawasi). Tujuan utama klasifikasi citra penginderaan jauh adalah untuk menghasilkan peta tematik, dimana suatu warna mewakili suatu objek tertentu. *Clustering* ini dapat dimanfaatkan untuk kompresi citra dengan menyimpan indeks tabel cluster dan rata-rata tiap cluster.

Secara lengkapnya algoritma dari *Improved Split and Merge Clustering* (ISMC) yaitu :

1. Penentuan nilai *threshold*

Terdapat 3 parameter *threshold* yang dibutuhkan oleh ISMC, yakni τ_m (*merge*), τ_s (*split*), dan Toleransi. Penghitungan τ_m dan τ_s adalah sebagai berikut :

$$\tau_s = SPLIT \times s \quad \text{dan} \quad \tau_m = MERGE \times s$$

Variabel *s* dihitung berdasarkan :

$$s = \beta \sum_{i=1}^n (\max_i - \min_i)^2$$

Dimana $\beta = 1.0 \times 10^{-4}$, *n* adalah jumlah komponen pada *vector* (*pixel*), \max_i dan \min_i adalah nilai maximum dan minimum pada elemen ke-*i* dari seluruh *vector*.

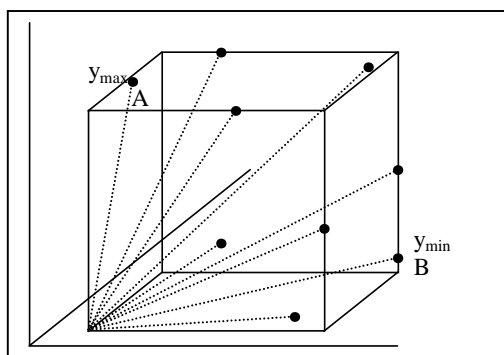
2. *Split cluster*

Urutan langkahnya sebagai berikut :

- Dilakukan pencarian nilai maximum dan minimum untuk tiap dimensi. Kemudian dilakukan *bounding*, sehingga semua *vector* berada dalam batasan tersebut.
- Pada Gambar 1 dicontohkan *feature space* yang terdiri dari 3 dimensi, yang berarti memiliki 2^3 sudut. Dari setiap sudut *boundary*, dilakukan pencarian *vector* terdekat dan terjauh.
- Selanjutnya tiap *vector* tersebut diukur jaraknya satu sama lain. Pasangan yang paling jauh, akan dijadikan sebagai kutub untuk membagi semua *vector*, ke dalam 2 *cluster* (proses *split*). Proses *split* baru boleh dilakukan, bila d_m antara A dan B tersebut melebihi *splitting threshold* τ_s . Bila tidak, maka proses langsung menuju langkah ke-3. Adapun nilai d_m ditentukan oleh rumus :

$$d_m = (y_{\max} - y_{\min})^T (y_{\max} - y_{\min})$$

- Proses *split* diharapkan akan menghasilkan 2 *cluster* dengan cara mengukur jarak seluruh *vector* pada *cluster* ini terhadap kedua *vector* kutub. Kutub yang terdekat itulah yang akan diikutinya sebagai *cluster* yang baru.
- Langkah ke-2 ini diulang untuk tiap *cluster*, hingga tidak terjadi *split* lagi.



Gambar 1. Contoh feature space 3 dimensi.

3. Penghapusan seluruh cluster yang tidak berelimen.

4. Penggabungan cluster

Tiap pasang cluster dihitung jaraknya terhadap cluster yang lain berdasarkan vektor mean.

Rumus jarak yang digunakan adalah :

$$d_m = (m_i - m_j)^T (m_i - m_j)$$

Bila d_m kurang dari merging threshold τ_m , maka lakukan penggabungan semua vector pada kedua cluster tersebut, sehingga menjadi 1 cluster.

Ulangi langkah ke-4 ini, hingga tidak ada lagi cluster yang bisa digabung.

5. Penentuan keanggotaan tiap vector

Rumus jarak berdasarkan Euclidean adalah :

$$d_k = (y - m_k)^T (y - m_k)$$

Vektor m_k adalah vektor mean cluster ke-k, sehingga untuk tiap pixel diukur jaraknya terhadap K cluster. Tiap vector diputuskan sebagai anggota cluster yang menghasilkan nilai d_k terkecil.

6. Penghitungan statistik cluster

Penghitungan ini meliputi trace dari matriks between-cluster scatter S_B , mean m , dan konvergensi. Rumus yang digunakan pada penghitungan S_B adalah :

$$S_B = \sum_{k=1}^K n_k (m_k - m)(m_k - m)^T$$

Dimana n_k adalah jumlah vector dalam cluster ke-k, sedangkan m adalah mean vector dari seluruh data set. Sedangkan rumus penghitungan mean vector untuk seluruh data set adalah :

$$m = \frac{1}{N} \sum_{k=1}^K n_k m_k$$

7. Pemeriksaan konvergensi

Rumus yang digunakan adalah :

$$\frac{Tr(S_B) - Tr(S_B^i)}{Tr(S_B)} < tol$$

S_B^i adalah S_B dari iterasi sebelumnya, sedangkan tol adalah nilai toleransi yang defaultnya

10

diset 0.05. Bila konvergensi terpenuhi, maka algoritma dihentikan, dan bila belum terpenuhi, maka ulangi langkah ke-2.

2.2. Reduksi Spektral dengan PCA

Metode PCA merupakan metode yang sering digunakan untuk mereduksi dimensi [4][6][12][15][16]. Adapun algoritma dari proses PCA adalah sebagai berikut :

- Penghitungan matrik kovarian
- Penghitungan eigenvalue dan eigenvector
- Menentukan matrik transformasi

Matrik kovarian dicari dengan rumus sebagai berikut:

$$C_x = \frac{1}{K-1} \sum_{j=1}^K (x_j - m)(x_j - m)^t$$

Lambang x_j menyatakan vektor pixel ke- j dari pixel sebanyak K . Sedangkan m sebagai rata-rata seluruh vektor dirumuskan sebagai berikut :

$$m = \frac{1}{K} \sum_{j=1}^K x_j$$

Proses selanjutnya adalah penghitungan eigenvalue dan eigenvector. Ide dasar dari PCA adalah mencari sistem koordinat baru dalam ruang ciri, dimana tiap ciri ditampilkan tanpa saling berkorelasi. Sebenarnya proses transformasi ini merupakan proses pemutaran sumbu hingga ditemukan sistem yang sesuai untuk syarat tersebut di atas.

$$y = G x$$

Dimana x adalah koordinat lama, G adalah matriks transformasi, dan y adalah koordinat baru hasil transformasi. Dalam koordinat baru ini diharapkan matriks kovarian y akan berbentuk matriks diagonal. Adapun rumus untuk matriks kovarian pada ruang y adalah :

$$C_y = \frac{1}{K-1} \sum_{j=1}^K (y_j - m_y)(y_j - m_y)^t$$

Variabel m_y adalah vektor mean dalam sistem koordinat y , sehingga dalam kaitannya dengan sistem koordinat x dapat ditunjukkan bahwa :

$$m_y = \frac{1}{K} \sum_{j=1}^K y_j = \frac{1}{K} \sum_{j=1}^K (Gx_j) = G \frac{1}{K} \sum_{j=1}^K x_j = Gm_x$$

Sedangkan m_x adalah vektor mean pada sistem koordinat x sehingga :

$$C_y = \frac{1}{K-1} \sum_{j=1}^K (Gx_j - Gm_x)(Gx_j - Gm_x)^t$$

$$C_y = G \left(\frac{1}{K-1} \sum_{j=1}^K (x_j - m_x)(x_j - m_x)^t \right) G^t$$

$$C_y = GC_x G^t$$

Sebagaimana telah dibahas, bahwa C_y merupakan matriks diagonal, oleh karena itu G dapat didefinisikan sebagai *transpose* dari *eigen vector* C_x dan G merupakan matriks diagonal yang elemennya merupakan *eigen value* dari C_x , yaitu :

$$C_y = \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \dots & \dots & \dots & 0 \\ 0 & 0 & \dots & \lambda_N \end{bmatrix}$$

Variabel N adalah dimensi rang ciri tersebut. Tiap elemen C_y menunjukkan variansi data *pixel* pada tiap komponen dalam sistem koordinat hasil transformasi. Nilai ciri (*eigen value*) ini dapat diurutkan secara *descending* menjadi $\lambda_1 > \lambda_2 > \dots > \lambda_N$, sedemikian hingga menunjukkan data *pixel* mencapai variansi maksimum pada komponen y_1 . Variansi terbesar kedua ditunjukkan oleh y_2 dan seterusnya, hingga variansi minimum di komponen y_N .

2.3. Encoding/Decoding

Pengkodean yang digunakan pada penelitian ini adalah metode *Lempel-Ziv Welch* (LZW). Metode *lossless compression* ini digunakan sebagai kombinasi untuk menghasilkan rasio kompresi yang lebih besar. Beberapa contoh dari *lossless compression* telah disebutkan pada sub bab sebelumnya. *Huffman coding* dan *Arithmetic coding* membutuhkan informasi tentang prioritas dari data masukan atau model statistik dari *source* [2]. Dalam beberapa kasus, sebuah *source model* yang cukup akurat sulit untuk mendapatkan pengkodeannya, khususnya ketika beberapa tipe data (seperti teks, grafik, dan gambar natural) dicampur.

Keunggulan dari LZW adalah metode ini tidak perlu tahu prioritas *source* atau menghitung statistik data masukan terlebih dahulu [2]. Metode LZW yang digunakan untuk kompresi data pertama kali dipublikasikan pada tahun 1977, kemudian algoritma ini disempurnakan oleh Terry Welch dan dipublikasikan kembali pada tahun 1984 [10][11]. Algoritma dari LZW ini sangat sederhana. Pada intinya, kompresi data LZW mengkodekan string dari karakter dengan kode tunggal. Setiap string baru

dari karakter yang ditemukan akan ditambahkan ke dalam table string. Algoritma ini tidak memperhatikan jenis data masukan. Semua data masukan akan dianggap sebagai karakter. Kompresi data terjadi pada saat algoritma ini menghasilkan output kode tunggal dari sebuah string karakter [11].

Kode yang dihasilkan algoritma LZW ini dapat mempunyai panjang yang berubah-ubah, tetapi harus memiliki lebih banyak bits daripada sebuah karakter tunggal. 256 kode yang pertama (jika menggunakan 8 bit karakter) adalah sebagai default yang digunakan untuk set standar karakter (kode ASCII). Kode sisanya akan digunakan untuk string hasil algoritma. Sebagai contoh untuk program yang menggunakan kode 12 bit, maka ini berarti kode 0-255 adalah individual bytes dan kode 256-4095 adalah substring.

3. METODOLOGI

Metodologi yang digunakan untuk penelitian ini dibagi menjadi 2 macam sebagaimana Tabel 1.

Tabel 1. Metode Kompresi

Metode 1	Citra di- <i>clustering</i> kemudian hasil <i>clustering</i> dikodekan dengan LZW
Metode 2	Citra direduksi menggunakan PCA, kemudian di- <i>clustering</i> , hasil dari <i>clustering</i> dan matrik transformasi spektral dikodekan dengan LZW

Pada proses reduksi spektral, dihasilkan matrik transformasi spektral dan citra tereduksi. Citra tereduksi ini kemudian di-*clustering*. Hasil dari clustering ini yang berupa indeks ke tabel cluster dan rata-rata tiap cluster dikodekan dengan menggunakan LZW bersama dengan matrik transformasi spektral.

Hasil akhir yang ditampilkan berupa rasio kompresi, waktu komputasi, jumlah cluster, dan ukuran kesalahan yang meliputi total *error*, maksimum *error*, dan rata-rata *error*.

4. UJI COBA

Data sample yang digunakan dalam uji coba ini terdiri atas 3 citra optik Landsat TM, dan 5 citra meteorologi GOES-8. Masing-masing data berformat bitmap dan memiliki tingkat keabuan 256 warna. Pada Tabel 2. diperlihatkan spesifikasi dari setiap data.

Data sampel dari Tabel 2 selanjutnya dieksekusi dengan tiga metode kompresi citra multispektral yang berbeda untuk mengetahui waktu komputasi, rasio kompresi, dan ukuran kesalahan yang dihasilkan.

Tabel 2. Sampel Data

Data	File Citra Multispektral	Satelit	Dimensi
1	Xlanier	Landsat TM	7
2	Surabaya	Landsat TM	6
3	Brazil	GOES-8	5
4	Florida	GOES-8	4
5	Claifornia	GOES-8	4
6	Argentina	GOES-8	4
7	Colorado	GOES-8	4
8	TMSby	Landsat TM	4

4.1. Waktu Eksekusi

Sub bab ini digunakan untuk mengetahui waktu yang dibutuhkan untuk proses kompresi pada masing-masing metode. Perbandingan waktu eksekusi pada masing-masing metode ini diperlihatkan pada Tabel 3.

Tabel 3. Waktu komputasi dalam detik.

	Metode 1	Metode 2
Data 1	9.13	0.84
Data 2	32.96	3.75
Data 3	5.21	2.65
Data 4	7.04	2.33
Data 5	5.91	3.76
Data 6	5.41	2.04
Data 7	0.9	0.51
Data 8	16.86	2.9

Dari data diatas, dapat diketahui bahwa waktu komputasi metode 2 lebih cepat dibandingkan metode 1. Perhitungan waktu komputasi ini juga dipengaruhi oleh beberapa faktor eksternal, yaitu besarnya memori, kecepatan processor, dan sistem operasi yang digunakan.

4.2. Rasio Kompresi

Rasio kompresi merupakan perbandingan antara ukuran citra sebelum kompresi dengan sesudah kompresi. Pada tabel 5.8 akan diperlihatkan perbandingan rasio kompresi pada tiap metode. Semakin kecil ukuran *file* yang dihasilkan proses kompresi, semakin besar nilai rasio kompresi yang didapatkan.

Tabel 4. Perbandingan rasio kompresi

	Metode 1	Metode 2
Data 1	7.38	8.82
Data 2	29.93	26.07
Data 3	37.75	33.58
Data 4	22.59	25.92
Data 5	15.76	16.68
Data 6	24.33	26.59
Data 7	8.9	9.08
Data 8	4.47	7.42

Ternyata dari data pada Tabel 4, metode 2 tidak selalu memiliki rasio kompresi yang terbesar dibandingkan dengan metode 1, yakni pada Data 2 dan Data 3. Namun demikian secara umum, nampak bahwa Metode 2 lebih efisien untuk mengkompresi.

4.3. Perubahan Nilai Threshold

Uji coba selanjutnya dilakukan dengan mengubah nilai *threshold* pada kedua metode. Perubahan ini dilakukan dengan tujuan untuk memperbanyak jumlah *cluster* yang dihasilkan proses ISMC dan diharapkan dapat mengurangi kesalahan setelah proses dekompresi. Tabel 5. memperlihatkan hasil perubahan kedua *threshold* pada ketiga metode. Sebagai contoh masukan adalah data 6 yaitu citra meteorologi GOES-8 Argentina.

Tabel 5. Pengaruh Perubahan threshold

Metode		Threshold	
		Merge = 50 Split = 750	Merge = 10 Split = 200
1	Jumlah Cluster	9	31
	Ukuran File	12520	25288
	Rasio Kompresi	21.28	10.54
	Waktu Komputasi	5.27	10.92
2	Jumlah Cluster	7	26
	Ukuran File	11526	23288
	Rasio Kompresi	23.12	11.44
	Waktu Komputasi	2.18	4.42

Dari data tersebut dapat diketahui bahwa perubahan *threshold* berpengaruh banyak pada hasil kompresi. Jumlah *cluster*, ukuran *file*, dan waktu komputasi menjadi lebih besar, sedangkan rasio kompresi menjadi lebih kecil.

4.4. Ukuran Kesalahan.

Sub bab ini digunakan untuk mengetahui ukuran kesalahan yang terjadi setelah proses dekompresi pada masing-masing metode. Pengukuran kesalahan dibagi menjadi tiga macam, antara lain total kesalahan, maksimum kesalahan, dan rata-rata kesalahan. Ukuran kesalahan tersebut dihitung berdasarkan *Euclidean Distance* [3] dengan rumus sebagai berikut :

$$D_i = \sqrt{\sum_{k=1}^p (x_{ik} - y_{ik})^2}$$

dimana x_i adalah *pixel* ke i pada citra hasil dekompresi, y_i adalah *pixel* ke i pada citra asli, dan p adalah jumlah *band*. Kesalahan yang ditimbulkan oleh proses kompresi/dekompresi dapat dilihat pada Tabel 6.

Tabel 6. Total kesalahan

Metode	Total Kesalahan	
	1	2
Data 1	1034314	1326352
Data 2	4160272	3171765
Data 3	3623046	3516164
Data 4	273840	260204
Data 5	2389086	2614673
Data 6	2286406	2426691
Data 7	497922	460821
Data 8	9617772	8795230

4.5. Contoh Citra Input dan Output

Contoh Citra input dapat dilihat pada Gambar 1 dan Gambar 2, yakni citra dari daerah Surabaya dari Band ke-5.



Gambar 2. Contoh Citra Input

Sedangkan citra output dari gambar di atas terdapat pada Gambar 3. Output yang dihasilkan oleh Metode 1 dan Metode 2 tersebut berdasarkan data input yang sama. Memang secara sepintas dengan kasat mata, tidak nampak perbedaan yang signifikan, namun bila digunakan analisa kuantitatif sebagaimana tersebut di atas, akan nampaklah perbedaan diantara keduanya.



Hasil Metode 1



Hasil Metode 2

Gambar 3. Contoh Citra Output

5. KESIMPULAN

1. Semakin sedikit jumlah cluster, semakin besar nilai rasio kompresi yang dihasilkan.
2. Dari segi waktu komputasi, metode 1 mempunyai waktu komputasi lebih lama dibandingkan metode 2. Sehingga dapat

dikatakan bahwa metode 2 dapat diunggulkan dari segi kecepatan.

3. Integrasi antara metode *clustering*, manipulasi spektral, serta pengkodean dapat digunakan secara efektif untuk mengkompresi citra penginderaan Jauh multispektral.
4. Manipulasi spektral yang digunakan yakni *Principal Component Analysis* (PCA) akan sangat mendukung kompresi ini, bila antar spektral memiliki korelasi yang tinggi. Hal ini sangat sesuai untuk citra yang berasal dari sensor optik.

DAFTAR PUSTAKA

- [1] Anil K. Jain, *Fundamentals of Digital Image Processing*, Prentice-Hall, Inc., Singapore, 1989.
- [2] Al Bovik-Department of Electrical and Computer Engineering The University of Texas at Austin, *Handbook of Image and Video Processing*, Austin-Texas, 2000.
- [3] Arto Kaarna, Pavel Zemcik, Heikki Kalviainen, and Jussi Parkkinen, Compression of Multispectral Remote Sensing Images Using Clustering and Spectral Reduction, *IEEE Trans. Geosci. Remote Sensing*, vol.38, no.2, March 2000.
- [4] Agus Zainal Arifin dan Aniati Murni, Algoritma Clustering adaptif untuk Klasifikasi Citra Penginderaan Jauh, *Prosiding Seminar Nasional Kecerdasan Komputasional II ICIS. vol. 2 No. 1. Universitas Indonesia*, 16 Oktober 2001.
- [5] James J. Simpson, Timothy J. McIntire, and Matthew Sienko, An Improved Hybrid Clustering Algorithm for Natural Scenes, *IEEE Trans. Geosci. Remote Sensing*, vol. 38, no. 2, March 2000.
- [6] John A. Richards, *Remote Sensing Digital Image Analysis*, Springer-Verlag Berlin Heidelberg, 1986.
- [7] Colm Mulcahy, Ph.D., Image Compression Using The Haar Wavelet Transform, www.cis.udel.edu/~amer/CISC651/, 1997
- [8] Alex Nicolau, A Wavelet Wading Pool, <http://www.cgl.uwaterloo.ca/~anicolao/wadingpool/WaveletWadingPool.html>, 1996.
- [9] Kristian Sandberg – University of Colorado, Haar Wavelet Transform, <http://amath.colorado.edu/courses/4720/2000Spr/Labs/Haar/haar.html>, April 2000.
- [10] David Marshall, Multimedia, *Module No: CM0340*, http://www.cs.cf.ac.uk/Dave/Multimedia/BSC_M_CALLER.html, 1999.

- [11] Mark Nelson, LZW Data Compression, *Dr. Dobb's Journal*, <http://dogma.net/markn/articles/lzw/lzw.htm>, Oktober 1989.
- [12] Agus Zainal Arifin, Algoritma Clustering Fuzzy Hibrida untuk Klsifikasi Citra Penginderaan Jauh, *Pra-Proseding Seminar Nasional Penginderaan Jauh dan Sistem Informasi Geografis, Universitas Gadjah Mada*, 7 April 2001.
- [13] Sani Muhamad Isa, Aniati Murni, Kompresi Citra Multispektral Menggunakan Karhunen-Loeve Transform Hybrid, *Proseding Seminar Nasional Kecerdasan Komputasional II ICIS. vol. 2 No. 1. Universitas Indonesia*, 16 Oktober 2001.
- [14] Mark Nelson, Jean-Loup Gailly, *The Data Compression Book*, New York, 1996
- [15] Agus Zainal Arifin dan Aniati Murni, Disain dan Implementasi Perangkat Lunak Klasifikasi Citra Inderaja Multispektral secara Unsupervised, *Jurnal Ilmu Komputer dan Teknologi Informasi*, Vol. 2, No. 1, Universitas Indonesia, Mei 2002.
- [16] Agus Zainal Arifin dan Wiwik Dyah Septiana Kurniati, Penggunaan Analisa Faktor untuk Klasifikasi Citra Penginderaan Jauh Multispektral, *Jurnal Ilmiah Teknologi Informasi*, Vol. 1, No. 1, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember, Agustus 2002.