

ALGORITMA CLUSTERING ADAPTIF UNTUK KLASIFIKASI CITRA PENGINDERAAN JAUH MULTISPEKTRAL

AGUS ZAINAL ARIFIN, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember, Surabaya
Sutorejo Prima Indah PZ-6 Surabaya, agusza@its-sby.edu

ANIATI MURNI, Fakultas Ilmu Komputer, Universitas Indonesia, Depok
Gedung Fasilkom UI Depok, aniat@caplin.cs.ui.ac.id

Abstrak-- Dalam penelitian ini, diusulkan sebuah algoritma *clustering* baru untuk melakukan klasifikasi secara *unsupervised* terhadap citra penginderaan jauh multispektral. Algoritma *Clustering* Adaptif (CA) ini mengintegrasikan metode *hierarchical* (*split* dan *merge*) dan *partitional clustering*. Proses *split* membagi dataset menjadi sejumlah *cluster* berdasarkan *histogram* dari komponen utama citra. Sedangkan proses *merge* berusaha menggabungkan secara ketat *cluster* yang telah dihasilkan, agar keberadaan *noise* tidak mengakibatkan *chain effect*. Adapun *partitional clustering* berusaha mendeteksi *prototype* tiap *cluster*, dengan memanfaatkan *Fuzzy C-Mean* (FCM). Algoritma CA ini diuji coba terhadap sejumlah citra penginderaan jauh dari satelit Landsat TM dan GOES-8. Untuk bahan perbandingan, sampel juga diuji coba dengan algoritma ISMC, yang berdasarkan penelitian sebelumnya telah terbukti lebih baik dari pada ISODATA dan SMC. Hasil perbandingan membuktikan, bahwa algoritma CA lebih unggul dalam hal kekompakan tiap *cluster* dan heterogenitas antar *cluster*. Keunggulan lainnya adalah lebih memudahkan penggunaannya, sebab hanya membutuhkan dua parameter, yakni *threshold merge* dan toleransi (untuk konvergensi FCM). Di samping itu, algoritma ini tidak meminta pengguna untuk menentukan jumlah *cluster* ataupun inisialisasi lokasi pusat tiap *cluster*.

Kata kunci : *clustering*, citra inderaja, *split*, *merge*, *fuzzy*.

I. PENDAHULUAN

Klasifikasi citra merupakan proses yang berusaha mengelompokkan seluruh *pixel* pada suatu citra ke dalam sejumlah *class* (kelas), sedemikian hingga tiap *class* merepresentasikan suatu entitas dengan properti yang spesifik [4][18]. Secara umum, algoritma klasifikasi dapat dibagi menjadi *supervised* (terawasi) dan *unsupervised* (tak terawasi) [8][18][19]. Pemilihannya bergantung pada ketersediaan data awal pada citra itu. Analisa *cluster* merupakan suatu bentuk pengenalan pola yang berkaitan dengan pembelajaran secara *unsupervised*, dimana jumlah pola kelas tidak diketahui [15][16]. Proses *clustering* melakukan pembagian *data set* dengan mengelompokkan seluruh *pixel* pada *feature space* (ruang ciri) ke dalam sejumlah *cluster* secara alami. Hampir semua algoritma *clustering* yang populer mengharuskan adanya inisialisasi *cluster* awal [6][12].

Padahal jumlah ini sangatlah sulit diketahui, sebab dibutuhkan seorang pakar yang menguasai konfigurasi objek tersebut. Fenomena ini mendorong para peneliti dalam bidang pengenalan pola (*pattern recognition*) untuk terus berusaha menghasilkan algoritma yang mampu mendeteksi jumlah *cluster* ini secara otomatis [6][12][15].

J. J. Simpson [15] telah mengembangkan algoritma *clustering*, yakni *Improved Split and Merge Classification* (ISMC) dengan menggabungkan proses *split* dan *merge*. Algoritma inilah, yang dalam tahap uji coba nanti, akan dijadikan sebagai pembanding, agar peningkatan kinerja pada algoritma CA dapat dilihat dengan mudah. Prosedur *split* pada ISMC ini berusaha membagi sebuah *cluster* secara *rekursif*. Pembagian sebuah *cluster* menjadi 2 *sub cluster* dilakukan berdasarkan pasangan *pixel* yang terjauh (2 *pixel* kutub). Tiap *cluster* yang terbentuk dipecah lagi dengan proses yang sama hingga ukurannya melampaui *threshold* untuk *split*. Kemudian dilakukan proses *merging* antar *cluster* yang berdekatan. Proses selanjutnya adalah *partitional*, yakni *assignment pixel* terhadap tiap pusat *cluster* untuk menentukan pusat *cluster* baru. Proses *split*, *merge*, dan *partitional* ini diulang hingga *konvergen*. Nampak bahwa mekanisme *split* pada algoritma tersebut tidak mempertimbangkan lokasi tempat berkumpulnya mayoritas *pixel*. Namun hanya mempertimbangkan jarak terjauh antar *pixel*. Hal ini bisa mengakibatkan pemotongan *cluster* yang berada di antara keduanya. Penyebabnya bisa berupa perbedaan distribusi atau ukuran *cluster* yang terlalu besar. Dengan demikian dibutuhkan metode *split* yang memperhatikan distribusi *pixel* dalam *feature space*. Distribusi ini dapat digambarkan melalui *histogram*, dimana tiap kurva yang terbentuk dapat diasosiasikan sebagai sebuah *cluster*.

Pembagian secara langsung ini juga dilakukan oleh Mehmet Celenk [9]. Metodenya menggabungkan *split* dan *merge*, dengan membagi seluruh citra menjadi *non-overlapping window* 4x4. Tiap *window* di-*split* menjadi 2 *cluster* dengan *K-means clustering*, dan ini bisa dilakukan secara paralel. Seluruh *cluster* hasil *split*, digabungkan dengan metode yang sama. Oleh karena tiap *window* memiliki 2 *cluster*, maka jumlah *cluster* menjadi sedemikian banyak, sehingga proses *merge* akan memakan waktu lama. Sehingga dibutuhkan metode yang cepat mungkin mampu membentuk sejumlah *cluster*.

Pada kenyataannya, proses pembentukan *cluster* dengan pencarian kurva pada *feature space* citra multispektral sangatlah sulit. Sebab dibutuhkan teknik *scanning* kurva yang sangat rumit. Cara yang termudah adalah mentransformasikannya menjadi satu dimensi, namun mampu mewakili seluruh spektrum. Proses ini biasa disebut dengan PCT (*Principal Component Transformation*) [3][5][14]. Bila proses *split* di atas menghasilkan *cluster* yang cukup banyak, maka dibutuhkan metode penggabungan yang lebih ketat. Penggabungan ini jangan sampai menyebabkan *chain effect*, bila pada citra tersebut terdapat *noise*. Sebab *noise* yang berderet diantara 2 *cluster* yang berjauhan dapat bertindak sebagai perantara.

Masalah lain yang harus dihadapi oleh algoritma *clustering* adalah ikut berperannya *noise* dalam penentuan *prototype cluster*. Salah satu metode yang dapat mengatasi kedua problema ini adalah *Fuzzy C-Means* (FCM) [7][11][6][12]. Dengan metode ini, *membership noise* terhadap semua pusat *cluster* tidak akan terlalu besar. Sehingga tidak akan terlalu menentukan lokasi pusat *cluster*. Namun demikian, FCM yang berbasis *partitional clustering* ini, tetap saja membutuhkan inisialisasi jumlah *cluster* dan *prototype* masing-masing *cluster*.

Penelitian ini bertujuan membangun sebuah sistem yang mampu melakukan *unsupervised classification* (klasifikasi tak terawasi) terhadap citra multispektral dengan lebih akurat, dalam artian kondisi *cluster* yang lebih kompak dan perbedaan antar *cluster* yang lebih meningkat, serta ketepatan pengenalan kelas yang lebih tinggi. Algoritma yang dikembangkan ini, diharapkan mampu melakukan *clustering* secara otomatis, yakni tanpa meminta pemakai untuk menentukan jumlah *cluster* dan posisi awal masing-masing pusatnya.

Adapun masalah yang dihadapi dapat dikategorikan menjadi 3 hal, yakni masalah *split*, *merge*, dan *partitional*. Masalah dalam proses *split* adalah pendeteksian lokasi terkonsentrasinya *pixel* agar *split* dapat dilakukan lebih cepat dan akurat. Sedangkan dalam hal *merge* adalah perlu dihindarinya *chain effect* sebagai akibat *noise*. Dalam hal *partitional*, perlu pendeteksian pusat *cluster* yang lebih akurat tanpa keterlibatan *noise* dalam penghitungannya.

II. ALGORITMA ISMC

Bab ini membahas algoritma *clustering* yang dijadikan sebagai pembanding, yakni algoritma *Improved Split and Merge Classification* (ISMC) [15]. Algoritma ISMC adalah hasil pengembangan dari SMC. Sedangkan SMC sendiri telah berhasil diaplikasikan untuk mengukur suhu permukaan laut berdasarkan kondisi awan di atasnya. Dalam uji cobanya, ISMC terbukti menghasilkan *cluster* yang lebih homogen, kondisi antar *cluster* yang lebih

heterogen, dan mampu mencapai konvergensi yang lebih cepat dari pada algoritma SMC dan ISODATA. Keunggulan lainnya adalah jumlah parameter inputnya yang lebih sedikit dan lebih mudah diisi, sebab terdapat rekomendasi yang dihitung berdasarkan sejumlah eksperimen. Berdasarkan berbagai keunggulan inilah, maka ISMC dijadikan sebagai algoritma pembanding.

Berikut ini adalah langkah-langkah algoritma ISMC :

1. Penentuan nilai *threshold*

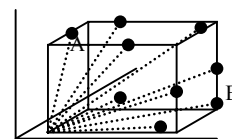
Parameter *threshold* ini adalah τ_m , τ_s , dan Toleransi. Nilai τ_m dan τ_s disesuaikan dengan karakteristik citra, yakni $\tau_s = \text{SPLIT} \times s$ dan $\tau_m = \text{MERGE} \times s$. Rekomendasi untuk $\text{SPLIT}=750$, $\text{MERGE}=50$, dan Toleransi=0,05. Sedangkan s dihitung berdasarkan :

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

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

2. *Split cluster*

Berdasarkan *component-wise*, dilakukan pencarian nilai maximum dan minimum untuk tiap dimensi. Sebagai *bounding*, sehingga semua *vector* berada dalam batasan tersebut. Pada Gambar 3.1 dicontohkan *feature space* 3 dimensi, yang berarti memiliki 2^3 sudut. Dari tiap sudut *boundary*, dicari *vector* yang terdekat dan terjauh.



Gambar 3.1. Contoh *feature space* 3 dimensi

Seluruh vektor yang terpilih, diukur jaraknya satu sama lain. Pasangan *vector* yang terjauh dijadikan sebagai kutub (pada Gambar 3.1, kutub adalah A dan B). Proses *split* baru boleh dilakukan, Bila jarak keduanya melebihi *splitting threshold* τ_s . Bila tidak, maka proses menuju langkah ke-3. Tiap *pixel* dikelompokkan ke salah satu kutub terdekat. Langkah ini diulang untuk tiap *cluster*, hingga tidak terjadi *split* lagi.

3. Penghapusan *cluster* yang tidak memiliki elemen.

4. Penggabungan *cluster*

Tiap pasang *cluster* dihitung jaraknya terhadap *cluster* lain berdasarkan vektor *mean*. Bila jaraknya kurang dari τ_m , maka kedua *cluster* itu digabungkan. Langkah ini diulang terus menerus, hingga tidak ada lagi *cluster* yang bisa digabung. Hasilnya adalah K buah *cluster*.

5. Penentuan keanggotaan tiap *vector*

Tiap *vector* ditetapkan sebagai anggota dari *cluster* yang terdekat diantara K buah *cluster* tersebut di atas.

6. Penghitungan S_B (*between-cluster scatter*) :

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

Vektor $m = \text{mean total cluster}$, vektor $m_k = \text{mean cluster } k$, dan $n_k = \text{jumlah anggota cluster } k$.

7. Pemeriksaan konvergensi

Konvergensi didasarkan pada tidak adanya perubahan signifikan pada $Tr(S_B)$ tersebut, rumusnya adalah :

$$\frac{Tr(S_B) - Tr(S'_B)}{Tr(S_B)} < tol \quad S'_B = S_B \text{ pada iterasi sebelumnya,}$$

sedangkan nilai *default tol* = 0,05.

Oleh karena jumlah *cluster* awal = 1, maka pada akhir iterasi pertama $S'_B = 0$. Idealnya, *trace* dari S_B atau $Tr(S_B)$ haruslah sebesar mungkin, untuk menyatakan bahwa seluruh *cluster* berbeda satu sama lain. Bila konvergensi terpenuhi, maka algoritma dihentikan, dan bila belum terpenuhi, maka ulangi langkah ke-2.

III. ALGORITMA CLUSTERING ADAPTIF

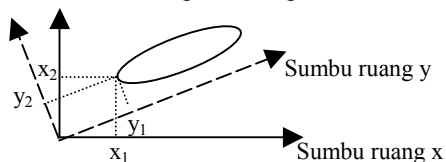
Nampak bahwa algoritma ISMC di atas membutuhkan 3 input parameter, yakni *threshold* untuk *split* dan *merge*, serta toleransi konvergensi. Adapun algoritma usulan, yakni algoritma *Clustering Adaptif* (CA) ini hanya meminta 2 parameter, yakni *threshold split* dan toleransi konvergensi. Proses selengkapannya adalah sebagai berikut :

1. Proses *split*, membagi secara langsung citra yang multidimensional tersebut menjadi sejumlah *cluster*.
2. Proses *merge*, menggabungkan tiap pasangan *cluster* yang jaraknya dianggap terlalu berdekatan. Metode yang digunakan adalah *complete link*. Output tahap ini adalah sejumlah vektor yang mewakili tiap *cluster*.
3. Proses *partitional* dengan *Fuzzy C Mean*, mendeteksi posisi vektor *mean* yang sebenarnya dari tiap *cluster*.

Tahap ke-3 yang dilakukan secara iteratif ini mengakhiri algoritma CA dengan menghasilkan nomor *cluster* untuk tiap *pixel*. Dengan demikian *pixel-pixel* yang memiliki kedekatan koordinat dalam *feature space*, akan dinyatakan sebagai anggota dari *cluster* yang sama.

III.1. Proses Split

Citra multispektral yang memiliki banyak dimensi itu, ditransformasikan dengan metode *Principle Component Transformation* (PCT). Proses ini sebenarnya adalah pemutaran sumbu sedemikian hingga variansi maksimum dicapai oleh komponen utama. Sebagaimana Gambar 3.2, sumbu ruang x dapat digantikan oleh sumbu ruang y, dengan variansi maksimum pada komponen utama.



Gambar 3.2. Ilustrasi modifikasi koordinat

Dengan demikian *array* multidimensi tersebut dapat ditransformasikan menjadi berdimensi tunggal. Distribusi *gray level array* ini divisualisasikan oleh histogram yang akan menampilkan sejumlah kurva. Tiap kurva ini dapat diasosiasikan sebagai satu *cluster*, sebagaimana penelitian

sebelumnya [8][14][16] yang telah melakukan hal yang sama namun dalam *feature space* 2 dimensi. Oleh karena perlu dicari interval batas tiap kurva, agar tiap *pixel* yang nilainya berada pada interval yang sama dapat digabungkan sebagai anggota suatu *cluster* yang sama pula. Selanjutnya, untuk tiap *cluster* yang terbentuk, perlu dicari vektor mean yang menjadi *prototype cluster* tersebut. Caranya dengan menghitung rata-rata dari seluruh vektor *pixel* yang menjadi anggotanya. Hasil akhir tahap ini adalah n vektor yang mewakili n *cluster*.

III.2. Proses Merge

Diantara tiap *cluster* yang dihasilkan oleh proses di atas sangat mungkin terdapat pasangan *cluster* yang sangat mirip, sehingga layak untuk digabung. Penggabungan (*merge*) ini harus dilakukan secara ketat, agar tiap *cluster* benar-benar kompak. Artinya, seluruh *pixel* dalam sebuah *cluster* yang sama, harus memiliki kemiripan yang tinggi satu sama lain. Selain itu, perlu dihindari terjadinya *chain effect*, yakni kondisi tergabungnya dua *cluster* yang sebenarnya tidak mirip, sebagai akibat terdapatnya *noise* yang menghubungkan keduanya. Metode yang diharapkan dapat mengakomodasi kebutuhan tersebut adalah *complete link*. Sedangkan ukuran kemiripan dihitung berdasarkan jarak *euclidean*.

Pada tahap awal, perlu dibuat tabel jarak antar *cluster* untuk mencari pasangan *cluster* yang paling mirip. Pasangan yang jaraknya paling dekat inilah, yang harus digabung. Selanjutnya tabel jarak antar *cluster* tersebut dihitung ulang, berkaitan dengan berkurangnya sebuah *cluster* ini. Jarak suatu *cluster* terhadap *cluster* baru ini ditentukan oleh jarak terdekatnya terhadap *cluster* baru tersebut. Proses penggabungan dan pengevaluasian tabel ini diulang terus menerus, hingga jarak terkecil yang ada telah melampaui *threshold merge*. Hasil akhir proses *merge* ini adalah vektor mean baru hasil penggabungan.

III.3. Proses Fuzzy C Mean

Tahap sebelumnya diasumsikan telah menghasilkan jumlah *cluster* yang optimal berdasarkan parameter yang diinputkan pengguna. Tahap *Fuzzy C Mean* ini berusaha mendeteksi pusat *cluster* yang sebenarnya, sehingga dapat menentukan *pixel-pixel* yang layak menjadi anggotanya. Pada setiap iterasi dilakukan penghitungan U_{ik} (membership *pixel* k terhadap *cluster* i) dan v_i (vektor mean dari *cluster* i), dengan rumus sebagai berikut :

$$U_{ik} = \left(\sum_{j=1}^c \left(\frac{D_{ik}}{D_{jk}} \right)^{\frac{2}{m-1}} \right)^{-1} \quad \forall i, k \quad \text{dan} \quad v_i = \frac{\sum_{k=1}^n U_{ik}^m X_k}{\sum_{k=1}^n U_{ik}^m} \quad \forall i$$

Proses yang sama akan dilakukan pada iterasi berikutnya hingga *konvergen*. Konvergensi tetap didasarkan pada tidak adanya perubahan signifikan pada $Tr(S_B)$.

IV. UJI COBA DAN EVALUASI

Uji coba algoritma CA dilakukan pada sampel citra inderaja multispektral yang telah diujicobakan untuk ISMC, agar peningkatan kinerjanya dapat dihitung. Dua dari tiga parameter ISMC yang digunakan oleh CA adalah *threshold merge* dan toleransi. Dengan demikian, untuk perbandingan berdasarkan kesamaan parameter, maka keduanya harus diisi nilai yang sama sebagaimana pada parameter ISMC. Perbandingan kedua adalah berdasarkan kesamaan jumlah *cluster* yang dihasilkan ISMC. Sehingga *threshold merge* CA dimodifikasi menjadi jumlah *cluster*. Dengan demikian *stopping criteria* proses *merge* sekarang adalah berdasarkan tercapainya jumlah *cluster* tersebut. Adapun kinerja yang lebih baik, ditentukan oleh lebih tingginya $Tr(S_B)$ dan lebih rendahnya $Tr(S_W)$ [15]. Bila vektor *mean K cluster* adalah m , vektor mean *cluster k* dengan jumlah anggota n_k adalah m_k , dan vektor *pixel i cluster k* adalah y_i^k , maka :

$$Tr(S_B) = \sum_{k=1}^K \sum_{i=1}^{n_k} \|y_i^k - m\|^2$$

$$Tr(S_W) = \sum_{k=1}^K n_k \|m_k - m\|^2$$

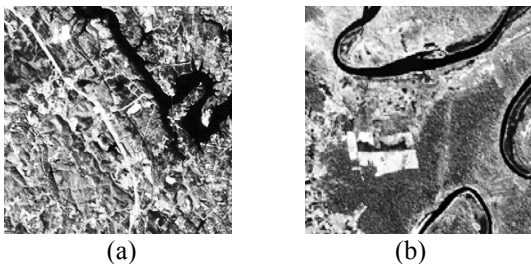
IV.1. Data sampel

Karakteristik data sampel dapat dilihat pada Tabel 4.1. Sampel dari Landsat TM sebenarnya adalah 7 band, oleh karena perbedaan resolusi, maka band 6 tidak digunakan, sehingga jumlahnya menjadi 6 band. Sedangkan pada GOES-8, dari 5 band tersedia, hanya band 2, 4, dan 5 yang digunakan, agar lebih relevan dengan uji coba ISMC [15] yang menggunakan AVHRR band 2, 3, dan 4.

Tabel IV.1. Sampel citra uji coba

Kode	Lokasi	Satelit	Ukuran
A	Jawa Tengah	Landsat TM	256 ²
B	Riau	Landsat TM	300 ²
C	California	GOES-8	256 ²
D	Galapagos	GOES-8	256 ²
E	Panama	GOES-8	250 ²
F	Texas	GOES-8	250 ²
G	Nicaragua	GOES-8	250 ²

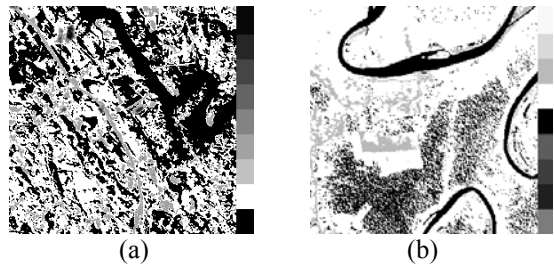
Gambar IV.1 menampilkan dua citra Landsat TM, yakni Jawa Tengah band 5 (a) dan Riau band 4 (b).



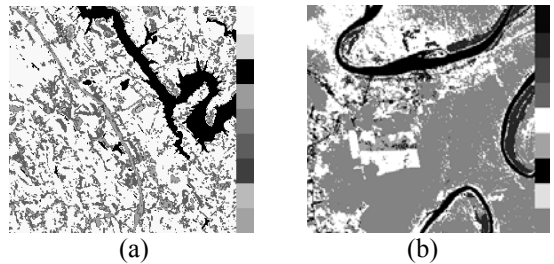
Gambar IV.1. Citra dari Sampel A dan B

IV.2. Eksekusi ISMC dan CA

Kualitas hasil ISMC dan CA dihitung berdasarkan $Tr(S_B)$ dan $Tr(S_W)$. Sesuai dengan rencana semula, uji coba terhadap algoritma CA dilakukan dua kali. Pertama berdasarkan kesamaan parameter dan kedua berdasarkan kesamaan jumlah *cluster*. Hasil klasifikasi ISMC ditunjukkan oleh Gambar IV.2. Sedangkan hasil uji coba CA berdasarkan kesamaan jumlah *cluster* ditunjukkan oleh Gambar IV.3. Keduanya terdiri dari Jawa Tengah (a) dan Riau (b) yang masing-masing memiliki 9 *cluster*. Uji coba berdasarkan kesamaan parameter tidak ikut ditampilkan, dengan alasan penghematan tempat.



Gambar IV.2. Hasil ISMC sampel A dan B



Gambar IV.3. Hasil CA sampel A dan B

Adapun Kualitas hasil ISMC dan CA dalam hal heterogenitas antar *cluster*, ditunjukkan oleh Tabel IV.2, dan dalam hal homogenitas ditunjukkan oleh Tabel IV.3.

Tabel IV.2. Nilai $Tr(S_B)$ sampel

Kode	$Tr(S_B)$		
	ISMC	CA Pertama	CA Kedua
A	4,478022e+07	5,382784e+07	4,757978e+07
B	4,401811e+07	8,321390e+07	7,608887e+07
C	2,773709e+08	2,898602e+08	2,878832e+08
D	3,611318e+08	3,740104e+08	3,657468e+08
E	6,924016e+07	7,156918e+07	7,062125e+07
F	1,732320e+08	1,747408e+08	1,712113e+08
G	2,060136e+08	2,174889e+08	2,135345e+08

Pada uji coba berdasarkan kesamaan parameter, nampak bahwa algoritma CA memiliki kelebihan dalam hal heterogenitas antar *cluster* yang berkisar antara 0,86 % (F) hingga 47,10 % (B). Kelebihan lainnya dalam hal homogenitas keanggotaan tiap *cluster* yang berkisar antara 32,99 % (F) hingga 79,43 % (F). Lama eksekusinya tidak

ditampilkan dalam tabel tersebut, sebab peningkatan kecepatan ternyata hanya terjadi pada sampel A (76,5 %), B (65,9 %), dan C (2,8 %), lainnya justru lebih lambat.

Tabel IV.3. Nilai $Tr(S_w)$ sampel

Ko de	$Tr(S_w)$		
	ISMC	CA Pertama	CA Kedua
A	2,052004e+07	8,617945e+06	1,444018e+07
B	3,950288e+07	8,126886e+06	1,523252e+07
C	2,461730e+07	9,319228e+06	1,631752e+07
D	1,725970e+07	4,020906e+06	1,966575e+07
E	4,741370e+06	1,922373e+06	3,917660e+06
F	8,211367e+06	5,502580e+06	8,948068e+06
G	1,486727e+07	3,354379e+06	1,179007e+07

Sedangkan pada uji coba berdasarkan kesamaan jumlah *cluster*, peningkatan heterogenitas berkisar antara 1,26 % (D) hingga 42,15 % (B). Namun sampel F justru mengalami penurunan -1,18 %. Dalam hal homogenitas, peningkatan berkisar antara 20,70 % (G) hingga 61,44 % (B). Namun penurunan juga terjadi, yakni pada sampel D (-13,94 %) dan F (-8,97 %). Kedua anomali ini sangat dipengaruhi oleh karakteristik citra itu sendiri yang biasa dikategorikan sebagai *uncertainty*.

V. KESIMPULAN

Berdasarkan uji coba ISMC dan CA dengan nilai parameter yang sama, ternyata terbukti bahwa algoritma CA mampu menghasilkan susunan *cluster* yang lebih kompak dan lebih menampakkan heterogenitas antar *cluster*. Terjadi peningkatan kecepatan sangat tajam pada sampel berdimensi tinggi, yang dalam hal ini Landsat TM. Proses *split* bisa dilakukan dengan cepat, bila mempertimbangkan juga faktor distribusi *gray level*. Penggunaan PCT untuk *histogram* sangat dipengaruhi oleh korelasi, oleh karena itu algoritma ini sangat sesuai untuk sampel yang tiap dimensinya saling berkorelasi.

V. REFERENSI

[1] Aniati Murni, *Metodologi Interpretasi Citra Inderaja Multitemporal dan Multisensor Berdasarkan Klasifikasi Uniform*, Disertasi : Program Pascasarjana Universitas Indonesia, 1997.

[2] Aniati Murni dan S. Setiawan, *Pengantar Pengolahan Citra*, Elex Media Komputindo, Jakarta, 1992.

[3] Anil K. Jain, Robert P. W. Duin, dan J. Mao, Statistical Pattern Recognition: A Review, *IEEE Transactions On Pattern And Machine Intelligence*, 22(1), 2000.

[4] Chein-I Chang dan H. Ren, An Experiment-Based Quantitative and Comparative Analysis of Target Detection and Image Classification Algorithms for

Hyperspectral Imagery, *IEEE Trans. on Geoscience and Remote Sensing*, 38(2), 2000.

[5] Chein-I Chang, Qian Du, Tzu-Lung Sun, dan Mark L. G. Althouse, A Joint Band Prioritization and Band-Decorrelation Approach to Band Selection for Hyperspectral Image Classification, *IEEE Trans. on Geoscience and Remote Sensing*, 37(6), 1999.

[6] H. Frigui dan R. Krishnapuram, A Robust Clustering Algorithm Based on Competitive Agglomeration and Soft Rejection of Outliers, *Proceedings of the 1996 Conference on Computer Vision and Pattern Recognition*, 1996.

[7] J.C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum, New York, 1981.

[8] John. A. Richards, *Remote Sensing Digital Image Analysis, An Introduction*, Springer-Verlag Berlin Heidelberg, 1986.

[9] Mehmet Celenk, Hierarchical color clustering for segmentation of textured images, *Proceedings of the 29th Southeastern Symposium on System Theory*, 1997.

[10] Morton Nadler dan Eric P. Smith, *Pattern Recognition Engineering*, John Wiley & Sons, New York, 1993.

[11] Nevin A. Mohamed, M. N. Ahmed dan A. A. Farag, Modified Fuzzy C-Mean in Medical Image Segmentation, *Proc. of IEEE-EMBS*, 20(3), 1998, 1377-1380.

[12] Nozha B., On Competitive Unsupervised Clustering, *Proceedings of the International Conference on Pattern Recognition (ICPR'00)*, Barcelona, 2000.

[13] Patrik B. G. Dammert, Jan I. H. Askne, dan S. Kuhlmann, Unsupervised Segmentation of Multitemporal Interferometric SAR Images, *IEEE Trans. on Geoscience and Remote Sensing*, 37(5), 1999.

[14] Ph. Schmid dan S. Fischer, Color segmentation for the analysis of Pigmented Skin Lesions, *Proceedings of the Sixth International Conference on Image Processing and Its Applications*, 2, 1997, 688-692.

[15] James J. Simpson, Timothy J. McIntire, dan Matthew Sienko, An Improved Hybrid Clustering Algorithm for Natural Scenes, *IEEE Trans. on Geoscience and Remote Sensing*, 38(2), 2000.

[16] R. M. Haralick dan L. G. Saphiro, *Computer and Robot Vision*, Vol. I dan II, Addison-Wesley, 1993.

[17] Rafael C. Gonzales dan Richard E. Woods, *Digital Image Processing*, Addison-Wesley Publishing Company, 1992.

[18] Richard O. Duda dan Peter E. Hart, *Pattern Classification and Scene Analysis*, John Wiley & Sons, New York, 1973.

[19] T. M. Lillesand dan R. W. Kiefer, *Remote Sensing and Image Interpretation*, John Wiley & Sons, 1994.

[20] William R. Dillon dan Matthew Goldstein, *Multivariate Analysis Methods and Applications*, John Wiley & Sons, New York, 1984.