

# PENGUNAAN ANALISA FAKTOR UNTUK KLASIFIKASI CITRA PENGINDERAAN JAUH MULTISPEKTRAL

Agus Zainal Arifin dan Wiwik Dyah Septiana Kurniati

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  
E-mail : agusza@its-sby.edu

## ABSTRAK

Proses *clustering* bisa berlangsung baik secara *hierarchical* (*split* dan *merge*) maupun *partitional* (partisi). Proses *split* yang pembagiannya berdasarkan *histogram* lebih mudah dilakukan pada satu dimensi, sehingga dibutuhkan proses transformasi. Metode transformasi yang umum digunakan adalah *Principal Component Analysis* (PCA). Namun PCA ternyata hanya didasarkan pada pencarian dimensi bervariansi maksimum, sehingga memungkinkan terjadinya *overlapping* kelas, dalam arti ada kelas yang tidak dapat dipisahkan.

Pada penelitian ini, metode transformasi yang digunakan adalah Analisa Faktor (*Factor Analysis* / *Canonical Analysis*). Metode ini lebih baik bila dibandingkan dengan metode *Principal Component Analysis* (PCA). Sebab, Analisa Faktor mentransformasi sekaligus memilah *cluster* dalam *feature space*. Tiga proses utama dalam penelitian ini yaitu *split*, *merge*, dan *partitional K-means clustering*. Citra multispektral ditransformasi menjadi satu dimensi. *Histogram* satu dimensi displit dengan pemilihan puncak kurva. *Merge* menggabungkan *cluster* hasil *split* tersebut. *Cluster* yang berdekatan digabungkan menjadi *cluster* baru. *K-means clustering* digunakan untuk mendeteksi lokasi pusat *cluster* (prototipe *cluster*) dan sekaligus mengelompokkan *pixel* ke setiap *cluster*.

Hasil penelitian ini dibandingkan dengan hasil algoritma *clustering* yang proses transformasinya menggunakan PCA. Hasil perbandingan membuktikan bahwa *clustering* yang proses transformasinya menggunakan Analisa Faktor menghasilkan heterogenitas antar *cluster* lebih tinggi ( $Tr(S_B)$  meningkat antara 0.83 % sampai 19.58 %). Adapun kekompakan tiap *cluster* tidak selalu optimal. Hal ini sangat mungkin disebabkan jumlah kelas sampel kurang banyak dan pengambilan sampel di tiap kelas kurang bervariasi.

Kata kunci : Analisa Faktor, *complete link*, *K-means clustering*, *Scatter within class*, *Scatter between class*

## I. Pendahuluan

Klasifikasi citra merupakan proses pengelompokan *pixel* pada suatu citra ke dalam sejumlah *class* (kelas), sehingga setiap kelas dapat menggambarkan suatu entitas dengan ciri-ciri tertentu [2][12][13][14]. Tujuan utama klasifikasi citra penginderaan jauh adalah untuk menghasilkan peta tematik, dimana suatu warna mewakili suatu objek tertentu. Contoh objek yang berkaitan dengan permukaan bumi antara lain air, hutan, sawah, kota, jalan, dan lain-lain. Sedangkan pada citra satelit meteorologi, proses klasifikasi dapat menghasilkan peta awan yang memperlihatkan distribusi awan di atas suatu wilayah.

Secara umum, algoritma klasifikasi dapat dibagi menjadi *supervised* (terawasi) dan *unsupervised* (tak terawasi) [2][12][13][14]. Pemilihannya bergantung pada ketersediaan data awal pada citra itu. Analisa *cluster* merupakan suatu bentuk pengenalan pola yang berkaitan dengan pembelajaran secara *unsupervised*, dimana

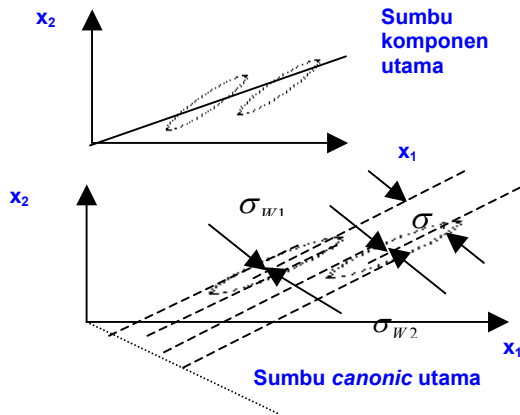
jumlah pola kelas tidak diketahui. Proses *clustering* melakukan pembagian *data set* dengan mengelompokkan seluruh *pixel* pada *feature space* (ruang ciri) ke dalam sejumlah *cluster* secara alami.

Metode *supervised* mengharuskan adanya *training set*. Akan tetapi *training set* untuk tiap kelas ini seringkali belum diketahui. Salah satu penyebabnya adalah sulitnya menentukan jumlah kelas yang sebenarnya terdapat pada citra itu disamping kesulitan untuk mencari lokasi-lokasi mana yang bisa dianggap paling mewakilinya. Fenomena ini mendorong para peneliti dalam bidang pengenalan pola (*pattern recognition*) untuk terus berusaha menghasilkan algoritma yang mampu mendeteksi jumlah *cluster* ini secara otomatis [11][12][13][14].

J. J. Simpson [11] telah mengembangkan algoritma *clustering*, yakni *Improved Split and Merge Classification* (ISMC) dengan menggabungkan proses

*split* dan *merge*. 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*.

Untuk mengatasi kelemahan di atas, maka pada penelitian selanjutnya [13][14] proses *split* diperbaiki dengan memperhatikan distribusi *pixel* dalam *feature space*. Distribusi ini dapat digambarkan melalui *histogram*, yang akan membentuk sejumlah kurva dimana tiap kurva yang terbentuk dapat diasosiasikan sebagai sebuah *cluster*[2][6]. Pada kenyataannya proses pencarian kurva pada *feature space* citra multispektral sangat sulit. Sebab dengan *feature space* yang berdimensi banyak, dibutuhkan teknik *scanning* kurva yang sangat teliti. Cara yang termudah adalah mentransformasikannya menjadi satu dimensi, namun mampu mewakili seluruh spektrum. Metode transformasi yang digunakan adalah *Principal Component Analysis* (PCA).



Gambar 1. Dua kelas yang tidak bisa dipisah oleh PCA  
Gambar 2. Dua kelas dipisahkan dengan Analisa Faktor  
Keterangan Gambar 2 :

$\sigma_A$  = Standar deviasi antar kelas (*standard deviation among the classes*)

$\sigma_{w1}$  = posisi *mean* dan penyebaran data dalam kelas 1

$\sigma_{w2}$  = posisi *mean* dan penyebaran data dalam kelas 2

Akan tetapi pada proses pemilihan puncak kurva *histogram*, diketahui bahwa metode PCA masih memungkinkan terjadinya *overlapping* kelas, dalam arti ada kelas yang tidak dapat dipisahkan, diperlihatkan dengan Gambar 1. Oleh karena itu dibutuhkan proses transformasi yang mampu mengatasi *overlapping* tersebut, yaitu proses transformasi dengan Analisa

Faktor. Analisa Faktor diharapkan mampu mentransformasi sekaligus menghasilkan *cluster* dengan optimal, dalam arti perbedaan antar kelas semakin besar dan kekompakan di dalam kelas semakin besar, karena metode transformasi ini dalam pencarian sumbu transformasi mempertimbangkan variansi maksimum dan keterpisahan kelas, diilustrasikan pada Gambar 2.

Permasalahan yang dihadapi dalam penelitian ini adalah :

1. Bagaimana melakukan *split* yang sekaligus memisahkan *cluster* dengan optimal.
2. Bagaimana melakukan *merge* (penggabungan) *cluster* yang berdekatan dengan ketat.
3. Bagaimana cara mendeteksi lokasi pusat *cluster*.

Adapun tujuan dari penelitian ini adalah mengklasifikasi citra penginderaan jauh multispektral dengan lebih akurat, dalam artian kondisi anggota dalam *cluster* lebih kompak dan perbedaan antar *cluster* yang lebih tinggi.

## II. Pengantar Klasifikasi

### II.1. Analisa Faktor

Analisa Faktor (*Factor Analysis* (FA)) digunakan untuk mereduksi dimensi.  $y = D^t x$  merupakan transformasi yang dibutuhkan untuk membentuk sumbu baru  $y$  yang mana kelas-kelas terpisah optimal. Langkah-langkah yang harus dilakukan dalam mencari sumbu transformasi :

- Dari setiap sampel dihitung mean ( $m$ ) dan kovarian ( $C_x$ )

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

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

Lambang  $x_j$  menyatakan vektor *pixel* ke- $j$  dari *pixel* sebanyak  $K$ .

- Matriks kovarian *within class* dirumuskan sebagai berikut :

$$C_w = \left\{ \sum_{i=1}^M (n_i - 1) C_i \right\} / S_n$$

Dimana  $C_i$  adalah matriks kovarian dari data pada kelas ke  $i$ ,  $M$  adalah jumlah total kelas,  $n_i$  adalah populasi dari kelas ke  $i$ , dan  $S_n$  adalah jumlah total *pixel* dari seluruh *training data* (sampel).

$$S_n = \sum_{i=1}^M n_i$$

- Matriks kovarian antar kelas (*among class covariance matrix*) dirumuskan sebagai berikut :

$$C_A = \xi \left\{ (m_i - m_0)(m_i - m_0)^t \right\}$$

Dimana  $m_i$  adalah *mean* dari kelas ke  $i$ ,  $\xi$  operator harapan (*expectation operator*), dan  $m_0$  adalah *global mean*.

- *Global mean* dihitung dengan persamaan berikut :

$$m_0 = \frac{\sum_{i=1}^M n_i m_i}{S_n}$$

Dimana  $m_0$  = *global mean*

$M$  = jumlah kelas

$m_i$  = *mean* kelas ke  $i$

$n_i$  = jumlah anggota kelas ke  $i$  (jumlah *pixel* dalam *training data* yang ke  $i$ )

$m_0$  = *global mean*

$S_n$  = jumlah total *pixel* dari seluruh *training data* (sampel).

- Persamaan *eigen value* dan *eigen vector*

$$(C_A - \lambda C_w)d = 0$$

Persamaan  $(C_A - \lambda C_w)d = 0$  bisa diubah ke bentuk persamaan  $(C_w^{-1}C_A - \lambda I)d = 0$  dengan serangkaian langkah berikut :

$$C_A d - \lambda C_w d = 0$$

$$C_A d = \lambda C_w d$$

$$C_w^{-1}C_A d = \lambda C_w^{-1}C_w d$$

$$C_w^{-1}C_A d = \lambda Id$$

$$C_w^{-1}C_A d - \lambda Id = 0$$

$$(C_w^{-1}C_A - \lambda I)d = 0$$

$A$  adalah matriks diagonal dari sekumpulan nilai *eigen* (*eigen value*) dan  $D$  adalah matriks dari vektor  $d$ .

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

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

## II.2. Pemilihan Puncak Kurva Histogram

*Feature space* pada citra *multispektral* atau *multidimensional*, dapat dibentuk menjadi sebuah *histogram* multidimensi. Bila yang digunakan adalah *histogram* 2 dimensi, maka setiap kelas akan nampak seperti sebuah bukit. Pendeteksian keberadaan suatu *cluster* melalui *histogram* ini dilakukan dengan menganalisa jumlah kurva beserta lokasinya masing-masing. Tiap puncak kurva diasosiasikan sebagai pusat dari suatu *cluster*. Dengan demikian, jumlah kurva dapat dinyatakan sebagai jumlah *cluster* [2][6].

*Histogram* multidimensi membutuhkan teknik *scanning* kurva yang rumit. Pada *histogram* 1 dimensi, proses pencarian ini dapat dilakukan secara linier. Proses deteksi terhadap keberadaan lembah hanya dilakukan searah, yakni dari satu dimensi saja. setiap *cluster* didefinisikan sebagai interval antara satu lembah dengan lembah yang lain. Dengan demikian pusat suatu *cluster* adalah rata-rata dari seluruh sampel yang membentuk kurva untuk *cluster* tersebut. Namun *histogram* 1 dimensi tidaklah merepresentasikan kondisi *data set* secara lengkap, sebab tidak semua dimensi terakomodasi. Oleh karena itu lokasi pusat masing-masing *cluster* ini hanyalah perkiraan awal, dan perlu dihitung ulang hingga konvergen.

## II.3. Complete Link

Algoritma *furthest-neighbor* (*complete link*), jarak antara suatu *cluster* dengan *cluster* baru ditentukan oleh jarak yang terjauh antara *cluster* ini dengan salah satu anggota *cluster* baru tersebut. Misalkan, terdapat sebuah *cluster* baru hasil gabungan dari *cluster* A dan B, maka jarak antara *cluster* baru ini ke *cluster* C, ditentukan dengan :

$$D_{(AB)C} = \max\{D_{AC}, D_{BC}\}$$

Sedangkan untuk *cluster* yang masing-masing adalah bentukan dari sejumlah *sub cluster*, maka rumusnya adalah sebagai berikut :

$$D = \max_{x \in X_i, x' \in X_j} \|x - x'\|$$

Bila terdapat ketentuan berupa *threshold*, maka algoritma ini akan nampak berusaha membentuk *complete sub graph* pada tiap *cluster*. Sebab jarak antara 2 *cluster* ditentukan oleh jarak *node* terjauh diantara keduanya. Bila jarak terjauh ini dianggap sebagai diameter *cluster*, maka diameter *cluster* pada tiap *level* tentu akan makin meningkat. Dan bila diameter ini melampaui *threshold*, maka penggabungan tidak akan dilakukan. Dengan demikian, dapat dijamin bahwa, semua *cluster* yang digabungkan adalah terbentuknya *cluster* yang cukup banyak.

Proses *merge* menggunakan algoritma *complete link* karena *complete link* ini dapat mengantisipasi kemungkinan terdapatnya *noise* dalam citra dan untuk membentuk *cluster* yang anggotanya sehomogen

mungkin [12][13]. Di samping itu penentuan *threshold* untuk *merge* yang sesuai sangat membantu menentukan jumlah akhir *cluster* yang dihasilkan.

## II.4. Partitional K-Means Clustering

Metode partisi ini melakukan *clustering* secara *iterative*, dan pada akhir iterasinya akan menghasilkan satu cara pengelompokan saja. Pada metode ini, suatu sampel yang sudah dialokasikan ke suatu *cluster*, pada iterasi berikutnya mungkin berpindah menjadi anggota *cluster* lain.

Algoritma ini membutuhkan inisialisasi jumlah *cluster* yang harus dihasilkan, beserta lokasi dari masing-masing pusat *cluster*. Tiap *pixel* diukur jaraknya terhadap setiap pusat *cluster* tersebut. Suatu *pixel* diputuskan menjadi anggota *cluster*  $i$ , bila jarak antara *pixel* tersebut dengan *cluster* ke- $i$  lebih dekat dibandingkan dengan *cluster* yang lain. Bila semua *pixel* sudah memilih *cluster*, maka dilakukan perhitungan pusat *cluster* yang baru. Caranya dengan mencari rata-rata dari seluruh *pixel* yang tergabung dengan *cluster* tersebut. Proses ini diulang terus menerus dan dihentikan dengan cara membatasi jumlah iterasinya atau bila semua pusat *cluster* sudah tidak mengalami perubahan.

Perhitungan jarak  $d(x, m_i)$  antara *pixel* ( $x$ ) dengan pusat *cluster* yang ke  $i$  ( $m_i$ ) menggunakan rumus berikut :

$$d(x, m_i)^2 = (x - m_i)'(x - m_i)$$

## III. Metodologi

Penggunaan Analisa Faktor untuk klasifikasi citra penginderaan jauh multispektral memiliki serangkaian proses sebelum menghasilkan *output* berupa citra yang sudah terklasifikasi. Langkah-langkah yang harus dilakukan meliputi:

### 3.1. Reduksi dimensi

Reduksi dimensi diawali dengan pengambilan data *pixel* citra penginderaan jauh dan disimpan dalam *array* matriks dua dimensi. User memilih *training data* untuk kelas yang diketahui (bisa dilakukan lebih dari sekali). Dari masing-masing *training data* dihitung *mean* dan kovariannya. Kemudian dilakukan perhitungan *global mean*, *covariance within class*, *covariance between class* dari *training data*. Selanjutnya dicari *eigen value*, *eigen vector*. Perhitungan nilai transformasi citra *input* ke sumbu *canonic* utama menghasilkan *array* satu dimensi dengan cara mengalikan *pixel* dari citra input multidimensi dengan *eigen vector* dari *eigen value* terbesar.

### 3.2. Split and merge clustering

Proses pemilihan puncak kurva *histogram*, meliputi pembuatan *histogram* dari *array* satu dimensi, proses reduksi *histogram*, dan pengelompokan nilai *pixel* yang dianggap sebagai *cluster-cluster* berdasarkan

algoritma *peak selection* (pemilihan puncak *histogram*).

*Merge* (penggabungan) *cluster* hasil *split*, mempunyai beberapa langkah, yaitu perhitungan *threshold merge*, perhitungan *mean* tiap *cluster*, perhitungan jarak *euclidian* antar *cluster* dari citra yang sudah tereduksi pada proses *peak selection* dan disimpan dalam tabel jarak. Proses *merge* atau penggabungan *cluster* berdasarkan tabel jarak *euclidian*. Langkah *merge* di atas diiterasi sampai *threshold merge* terlampaui oleh jarak *euclidian* terkecil. Hasilnya berupa jumlah *cluster* dan *mean* tiap *cluster*.

### 3.3. Partitional Clustering

Langkah yang dilakukan pada proses *partitional K-means Clustering* adalah pembuatan tabel *euclidian* dalam setiap iterasi untuk menghitung jarak antar *cluster*, perhitungan kembali *mean* tiap *cluster* yang nilainya selalu berubah, proses *pixel assignment* ke *cluster* terdekat berdasarkan perhitungan jarak *euclidian*, perhitungan *trace*  $S_B / Tr(S_B)$  dimana  $S_B$  adalah *scatter between class*, yang digunakan sebagai analisa data proses klasifikasi. Proses *partitional* diiterasi terus sampai konvergen. Setelah proses *partitional*, maka dilakukan perhitungan *scatter within class* ( $S_W$ ) untuk analisa. Hasil akhir berupa matriks dua dimensi yang isinya *pixel* yang sudah berkelompok berdasarkan kelas-kelas. Hasil klasifikasi ditampilkan dengan menuliskannya ke *bitmap*. Proses pewarnaan kelas berdasarkan nilai warna yang disimpan dalam file warna, sehingga setiap kelas mempunyai satu warna.

## IV. Uji Coba

Data sampel yang digunakan dalam uji coba ini terdiri dari beberapa citra optik Landsat TM dan GOES-8 dan dapat dilihat pada Tabel 1.

Tabel 1. Sampel citra untuk uji coba

Kode	Lokasi	Satelit	Ukuran	Band
A	Jawa Tengah	Landsat TM	256 <sup>2</sup>	6
B	Riau	Landsat TM	300 <sup>2</sup>	6
C	California	GOES-8	256 <sup>2</sup>	3
D	Galapagos	GOES-8	256 <sup>2</sup>	3
E	Panama	GOES-8	250 <sup>2</sup>	3
F	Texas	GOES-8	250 <sup>2</sup>	3
G	Nicaragua	GOES-8	250 <sup>2</sup>	3
H	SurabayaTM	Landsat TM	256 <sup>2</sup>	6

Sebagai pembanding digunakan *Algoritma Clustering Adaptif untuk Klasifikasi Citra Penginderaan Jauh Multispektral* karena metode transformasi yang digunakan adalah *Principal Component Analysis* (PCA) [12]. Untuk selanjutnya akan disebut sebagai algoritma *clustering* dengan metode transformasi PCA.  $S_B$  dari metode ini disebut  $S_B$  PCA dan  $S_W$ -nya disebut  $S_W$  PCA.

Hasil eksekusi pada penelitian ini,  $S_B$ -nya disebut  $S_B$  Analisa Faktor dan  $S_W$ -nya disebut  $S_W$  Analisa

Faktor. Uji coba dilakukan 2 kali, yaitu berdasarkan kesamaan *threshold merge* dan kesamaan jumlah kelas.

Ada 2 faktor yang digunakan dalam analisa, yaitu *trace*  $S_B$  ( $Tr(S_B)$ ) dan *trace*  $S_W$  ( $Tr(S_W)$ ). Peningkatan yang terjadi pada  $Tr(S_B)$  menunjukkan adanya peningkatan perbedaan antar *cluster*. Idealnya,  $Tr(S_B)$  haruslah setinggi mungkin, sebab perbedaan antar *cluster* harus sebesar mungkin.

Sedangkan penurunan yang terjadi pada *trace* matriks *within cluster scatter matriks*  $Tr(S_W)$  menunjukkan adanya peningkatan kekompakan pada masing-masing *cluster*. Idealnya  $Tr(S_W)$  haruslah serendah mungkin atau dengan kata lain keadaan tiap data dalam setiap *cluster* haruslah sehomogen mungkin. Scatter between class ( $S_B$ ) dari total kelas sebanyak  $C$  dengan  $N_i$  adalah jumlah anggota kelas ke  $i$ ,  $m_i$  adalah mean kelas ke  $i$ , dan  $m_0$  adalah *global mean*.

$$S_B = \sum_{i=1}^C N_i (m_i - m_0)(m_i - m_0)^t$$

Scatter within class ( $S_W$ ) dari total kelas sebanyak  $C$  dengan  $x_{ij}$  adalah pixel ke  $j$  dari kelas ke  $i$ , dan  $m_i$  adalah mean kelas ke  $i$ .

$$S_W = \sum_{i=1}^C \sum_{x_{ij} \in \omega_i} (x_{ij} - m_i)(x_{ij} - m_i)^t$$

#### IV.1. Berdasarkan kesamaan threshold merge

Dari Tabel 2 dapat dilihat penggunaan Analisa Faktor untuk klasifikasi citra penginderaan jauh multispektral menghasilkan  $Tr(S_B)$  yang lebih besar dibandingkan dengan algoritma pembandingan yang transformasinya menggunakan PCA, yaitu antara 1.27% sampai 19.58%. Hal ini berarti perbedaan antar kelas meningkat.

Tabel 2. Perbandingan  $Tr(S_B)$

Kode	$S_B$ PCA	$S_B$ Analisa Faktor	Kenaikan
A	53.827.840	66.929.360	19,58 %
B	83.213.900	86.684.100	4,00 %
C	289.860.200	309.751.900	6,42 %
D	374.010.400	378.700.300	1,27 %
E	71.569.180	76.471.810	6,41 %
F	174.740.800	189.999.600	8,03 %
G	217.488.900	220.774.100	1,49 %

Tabel 3. Perbandingan  $Tr(S_W)$

Kode	$S_W$ PCA	$S_W$ Analisa Faktor	Penurunan
A	8.617.945	3.588.780	58,37 %
B	8.126.886	6.130.480	24,57 %
C	9.319.228	5.839.713	37,34 %
D	4.020.906	5.772.958	-39,44 %
E	1.922.373	1.917.480	0,25 %
F	5.502.580	3.009.024	45,32 %
G	3.354.379	4.230.041	-26,71 %

Tabel 3 memperlihatkan pada penggunaan Analisa Faktor untuk klasifikasi, sebagian sampel yaitu

A, B, C, E dan F,  $Tr(S_W)$  lebih kecil yaitu antara 0,25% sampai 58,37%. Hal ini berarti homogenitas anggota dalam sebuah *cluster* meningkat. Sedangkan pada sampel D dan G,  $Tr(S_W)$  justru lebih besar dari algoritma pembandingan, sehingga penurunan dikatakan -39,44% dan -26,71%. Hal ini sangat mungkin disebabkan :

- Jumlah kelas sampel kurang banyak.
- Pengambilan sampel di tiap kelas kurang bervariasi sehingga sampel kurang lengkap.
- Dalam sampel yang diambil terdapat *noise / outlier*

#### IV.2. Berdasarkan kesamaan jumlah kelas

Uji coba berdasarkan jumlah kelas ditujukan untuk memaksa program menghasilkan jumlah *cluster* yang sama dengan jumlah yang dihasilkan oleh algoritma pembandingan [12]. Hal ini dilakukan agar dapat diperoleh hasil perbandingan yang lebih obyektif. Sebab meskipun inisialisasi variabel *merge* sama, akan tetapi menghasilkan jumlah *cluster* atau kelas yang berbeda. Pemaksaan untuk menghasilkan jumlah *cluster* yang sama dengan algoritma pembandingan bisa dilakukan dengan mengganti kriteria berhenti pada tahap *complete link*. Kriteria tersebut diganti dengan persyaratan bila jumlah *cluster* sudah mencapai batas yang ditentukan, maka proses penggabungan dihentikan.

Tabel 4. Perbandingan  $Tr(S_B)$

Kode	$S_B$ PCA	$S_B$ Analisa Faktor	Kenaikan
A	53.827.840	65.107.270	17,32 %
B	83.213.900	85.616.080	2,81 %
C	289.860.200	307.816.400	5,83 %
D	374.010.400	373.377.800	0,83 %
E	71.569.180	76.511.100	6,46 %
F	174.740.800	187.851.700	6,98 %
G	217.488.900	220.132.700	1,20 %

Tabel 4 menunjukkan ketika program dipaksa untuk menghasilkan kelas yang sama,  $Tr(S_B)$  tetap lebih baik dari algoritma pembandingan, berarti perbedaan antar *cluster* meningkat, antara 0,83% sampai 17,32%.

Tabel 5. Perbandingan  $Tr(S_W)$

Kode	$S_W$ PCA	$S_W$ Analisa Faktor	Penurunan
A	8.617.945	5.181.354	39,88 %
B	8.126.886	7.056.933	13,17 %
C	9.319.228	7.293.623	21,74 %
D	4.020.906	10.868.380	-75,83 %
E	1.922.373	1.838.025	4,39 %
F	5.502.580	4.516.709	17,92 %
G	3.354.379	4.710.307	-40,4 %

Tabel 5 memperlihatkan sebagian sampel yaitu A, B, C, E dan F  $Tr(S_W)$  lebih kecil antara 4,39% sampai 39,88% meskipun program dipaksa menghasilkan jumlah kelas yang sama dengan algoritma pembandingan. Sedangkan pada sampel D dan G,  $Tr(S_W)$  lebih besar. Hal ini seperti yang telah dievaluasi pada sub bab IV.1. tentang perbandingan  $Tr(S_W)$ .

### IV.3. Perbandingan Waktu dan Jumlah Iterasi Konvergen

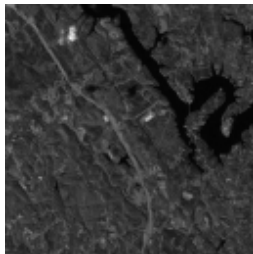
Dari Tabel 6 dan 7 dapat dilihat bahwasannya :

- Jumlah waktu yang dibutuhkan untuk eksekusi program (kolom “Waktu (dt)”) pada sampel A, B, H lebih besar dibandingkan dengan sampel C, D, E, F, G. Hal ini berkaitan dengan jumlah *band*. Semakin banyak jumlah *band* yang terlibat dalam uji coba maka semakin besar waktu yang dibutuhkan.
- Jumlah iterasi yang dibutuhkan untuk mencapai konvergen pada proses *K-means clustering* (kolom “Iterasi”) dari tiap sampel berbeda-beda.

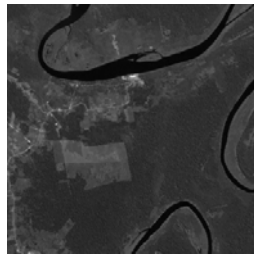
Tabel 6. Hasil dengan kesamaan threshold merge

Kode	Cluster	Iterasi	Waktu (dt)
A	42	22	205,48
B	26	19	140,97
C	36	13	66,96
D	25	20	47,64
E	19	18	37,98
F	43	8	41,89
G	29	21	57,86
H	28	21	140,50

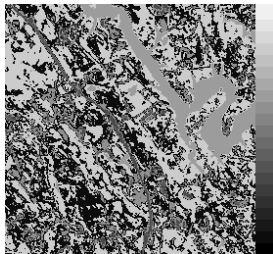
### IV.5. Contoh Input dan Output



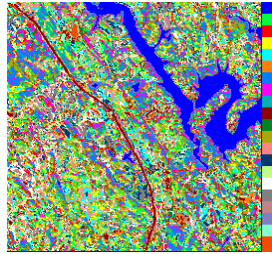
Sampel A band 5



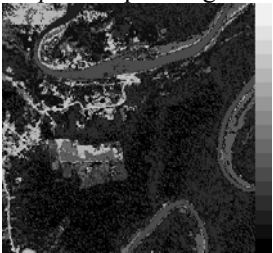
Sampel B band 5



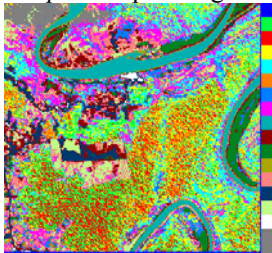
Output sampel A dg PCA



Output sampel A dg FA



Output sampel B dg PCA



Output sampel B dg FA

Gambar 3. Input dan Output

FA adalah singkatan dari *Factor Analysis* (Analisa Faktor).

Tabel 7. Hasil dengan kesamaan jumlah kelas

Kode	Cluster	Iterasi	Waktu (dt)
A	21	24	128,34
B	17	23	131,81
C	19	19	48,89
D	17	21	45,63
E	15	17	33,05
F	18	10	26,03
G	17	21	44,95

### V. Kesimpulan

Dari beberapa hasil uji coba yang telah dilakukan, didapatkan beberapa kesimpulan sebagai berikut :

1. Penggunaan Analisa Faktor untuk klasifikasi citra penginderaan jauh multispektral lebih mampu memisahkan *cluster* dibandingkan algoritma *clustering* yang transformasinya menggunakan Analisa Komponen Utama (*Principal Component Analysis*). Hal ini bisa dilihat dengan meningkatnya  $Tr(S_B)$  dari 0,83% sampai 19,58%, sebab Analisa Faktor mentransformasi sekaligus memisahkan *cluster* dalam *feature space*.
2. Dari 7 sampel yang digunakan dalam uji coba, homogenitas anggota *cluster* dari 5 sampel mengalami peningkatan, antara 0,25% sampai 58,37%. Sedangkan pada 2 sampel, homogenitas anggota *cluster* mengalami penurunan. Hal ini dapat dipahami karena Analisa Faktor lebih menekankan pada heterogenitas antar kelas.
3. Algoritma *clustering* yang memberikan *user* kesempatan untuk ikut memberikan pengetahuan berupa *training set* sangat membantu dibandingkan tanpa melibatkan pengetahuan *user*.
4. Pada *Kmeans clustering*, ternyata kecepatan menuju konvergensi pada tiap sampel tidak sama, hal ini sangat dipengaruhi representasi datanya.
5. Penambahan jumlah *band* yang terlibat dalam proses *clustering*, akan memakan waktu yang jauh lebih lama, sebab tiap *pixel* akan menjadi vektor yang dimensinya lebih banyak. Hal ini ditunjukkan oleh perbedaan waktu eksekusi pada citra Landsat TM 5 band dan citra GOES-8 3 band.

### DAFTAR PUSTAKA

- [1] William R. Dillon dan Matthew Goldstein, *Multivariate Analysis Methods and Applications*, John Wiley & Sons, New York, 1984.
- [2] John. A. Richards, *Remote Sensing Digital Image Analysis, An Introduction*, Springer-Verlag Berlin Heidelberg, 1986.
- [3] Scalkoff, Robert J., *Digital Image Processing and Computer Vision*, John Wiley & Sons, 1989
- [4] Low, Andrian, *Introductory Computer Vision and Image Processing*, McGraw-Hill Book Company, 1991

- [5] Rafael C Gonzales dan Richard E. Woods, *Digital Image Processing*, Addison-Wesley Publishing Company, 1992.
- [6] T.M. Lillesand, R. W. Kiefer, *Remote Sensing and Image Interpretation*, John Wiley & Sons, 1994.
- [7] *Erdas® Imagine Tour Guides<sup>TM</sup>*, Erdas®, Inc., Atlanta, Georgia, 1997.
- [8] *Erdas® Field Guide<sup>TM</sup>*, Fourth Edition, Erdas®, Inc., Atlanta, Georgia, 1997.
- [9] Te-Ming Tu, Chin-Hsing Chen, Jiunn-Lin Wu, dan Chein-I Chang, *A Fast Two-Stage Classification Method for High-Dimensional Remote Sensing Data*, IEEE Transactions on Geoscience and Remote Sensing, Vol. 36, No.1, Januari 1998.
- [10] 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 Transactions on Geoscience and Remote Sensing, Vol. 37, No.6, November 1999.
- [11] James J. Simpson, Timothy J McIntire, dan Matthew Sienko, *An Improved Hybrid Clustering Algorithm for Natural Scenes*, IEEE Transactions on Geoscience and Remote Sensing, Vol. 38, No.2, Maret 2000.
- [12] Agus Zainal Arifin, *Algoritma Clustering Fuzzy Hibrida untuk Klasifikasi Citra Penginderaan jauh*, Pra-Proseding Seminar Nasional Penginderaan Jauh dan Sistem Informasi Geografis, Universitas Gadjah Mada, 7 April 2001.
- [13] Agus Zainal Arifin dan Aniati Murni, *Algoritma Clustering Adaptif untuk Klasifikasi Citra Penginderaan Jauh Multispektral*, Proseding Seminar Nasional Kecerdasan Komputasional II ICIS, Vol. 2, No. 1, Universitas Indonesia, 16 Oktober 2001.
- [14] 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.