

Implementasi Fuzzy C-Means untuk Pengelompokan Citra Batik Berdasarkan Motif dengan Fitur Tekstur

Alvian A. Pratama, Nanik Suciati, Diana Purwitasari

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

Jl. Arief Rahman Hakim, Surabaya 60111

E-mail: nanik@if.its.ac.id

Abstrak— Batik Indonesia oleh UNESCO telah ditetapkan sebagai Warisan Kemanusiaan untuk Budaya Lisan dan Nonbendawi sejak 2 Oktober, 2009. Akan tetapi, banyak masyarakat Indonesia sendiri yang masih belum familier dengan jenis-jenis batik. Oleh karena itu, dalam tugas akhir ini diimplementasikan sistem yang bisa digunakan untuk membantu mengelompokkan jenis-jenis batik. Proses pengelompokkan citra membutuhkan fitur citra untuk dibandingkan. Di dalam tugas akhir ini, fitur citra yang digunakan untuk mengenali tekstur batik didapatkan melalui beberapa metode. Metode-metode tersebut adalah *Discrete Wavelet Transform (DWT)*, *Rotated Wavelet Filter (RWF)*, dan *Grey Level Co-occurrence Matrix (GLCM)*. Kemudian, hasil ekstraksi akan dikelompokkan ke dalam motifnya dengan menggunakan metode pengklasteran *Fuzzy C-Means (FCM)*. Penggunaan FCM untuk memberikan label pada batik yang memiliki lebih dari satu motif memberikan hasil yang cukup akurat. Sayangnya hasil pengelompokkan secara umum ternyata tidak terlalu baik. Hal ini kemungkinan besar dikarenakan banyaknya terjadi penghitungan nilai optimum lokal sehingga iterasi proses pengklasteran terhenti sebelum waktunya. Di sisi lain, penggunaan fitur tambahan RWF dan GLCM untuk melengkapi DWT mampu memberikan hasil pengenalan motif yang lebih baik. Dari percobaan yang dilakukan, nilai *Overall F-Measure* tertinggi didapatkan saat penggunaan *fuzziness* 2 yaitu sebesar 0,364. Hal ini menunjukkan bahwa pemilihan nilai *fuzziness* yang tepat diperlukan untuk mendapatkan hasil pengklasteran yang baik.

Kata Kunci : *Discrete Wavelet Transform, Fuzzy C-Means Clustering, Grey Level Cooccurrence Matrix, Pengenalan motif batik, Rotated Wavelet Filter*

I. PENDAHULUAN

BATIK Indonesia, sebagai warisan budaya telah memiliki sejarah yang cukup panjang. Sehingga saat ini berbagai jenis batik saling mempengaruhi satu sama lain yang mengakibatkan adanya beberapa jenis batik yang memiliki lebih dari satu motif (biasanya di sebut batik *multi-label*). Hal ini terkadang menyulitkan proses pengklasifikasian yang ingin dilakukan.

Sebelumnya telah dilakukan tugas akhir yang membuat sistem untuk pengklasifikasian batik. Dalam tugas akhir tersebut digunakan *Rotated Wavelet Filter* untuk mengenali tekstur batik dan *Neural Network* untuk pengklasifikasiannya [1]. Yang membedakan dengan penelitian tersebut adalah,

dalam tugas akhir ini akan digunakan beberapa metode untuk proses pengenalan tekstur. Metode-metode tersebut adalah *Grey Level Co-occurrence Matrix (GLCM)*, *Discrete Wavelet Transform (DWT)*, dan *Rotated Wavelet Filter (RWF)*. Sedangkan proses pengklasteran akan menggunakan *Fuzzy C-Means (FCM)*. Dengan menggunakan FCM, hasil pengenalan pola batik bisa memberikan lebih dari satu motif, sehingga bisa diaplikasikan untuk batik *multi-label*.

II. METODE PENELITIAN

Secara garis besar, terdapat tiga tahap utama yang perlu dilakukan dalam sebuah proses pengklasteran. Pertama adalah mengambil objek *input*. Kedua adalah proses ekstraksi fitur sebagai representasi objek. Tahap terakhir adalah proses pengklasteran itu sendiri.

Citra batik yang dijadikan objek dalam penelitian ini didapatkan dari buku Batik: Pengaruh Zaman dan Lingkungan karangan Santosa Doellah [2]. Citra tersebut berupa citra berwarna sehingga diperlukan proses *greyscaling* supaya proses ekstraksi fitur bisa berjalan dengan baik. Proses persiapan suatu citra untuk proses ekstraksi fitur seperti ini biasanya disebut preproses. Setelah preproses selesai tahap ekstraksi fitur bisa dilakukan. Proses-proses ekstraksi fitur tersebut dijelaskan di sub-bab berikut ini.

A. Grey Level Cooccurrence Matrix

Dalam suatu tekstur, distribusi *grey-level pixel* bisa dideskripsikan sebagai statistik, contohnya kemungkinan dua *pixel* yang berdekatan memiliki *grey-level* yang sama. Informasi *grey-level* ini bisa dijadikan matriks dua dimensi yang disebut dengan *Grey Level Cooccurrence Matrix*. GLCM dihitung berdasarkan berapa sering suatu *pixel* dengan *grey-level* i muncul di horisontal, vertikal, atau diagonal berpasangan dengan *pixel* dengan *grey-level* j .

Beberapa fitur yang bisa diekstraksi dari GLCM adalah energi, kontras, homogenitas, dan korelasi. Energi menghitung total tiap elemen pangkat dua. Kontras menghitung variasi lokal *grey-level* dalam GLCM. Korelasi menghitung kemungkinan kemunculan secara bersamaan sepasang *pixel*. Homogenitas menunjukkan jarak distribusi elemen-elemen dalam GLCM. Dalam suatu tekstur biasanya ada homogenitas yang berulang-ulang, sehingga fitur ini paling cocok digunakan untuk klasifikasi tekstur [4].

Untuk Homogenitas dari GLCM, dihitung untuk empat arah

($\theta = 0^\circ, 45^\circ, 90^\circ$, dan 135°). Vektor fitur dengan ukuran 4 dibuat untuk masing-masing citra. Homogenitas dihitung berdasarkan persamaan:

$$\text{Homogenitas} = \sum_{i,j} \frac{p(i,j)}{1 + |i - j|} \quad (1)$$

B. Discrete Wavelet Transform

Wavelet adalah sebuah fungsi matematika yang digunakan untuk merepresentasikan sinyal waktu yang kontinu ke dalam skala lain. Dengan *wavelet* dapat dilakukan analisis terhadap fungsi baik pada domain waktu maupun frekuensi resolusi yang berbeda [5]. Salah satu jenis *wavelet* yang terkenal adalah *wavelet* Daubachies. *Wavelet* ini banyak digunakan untuk mendefinisikan ruang multiresolusi.

Untuk kasus sinyal pada dimensi 2-D biasanya dilakukan dengan menerapkan struktur bank filter secara terpisah terhadap sinyal citra. Digunakan Low-Pass Filter atau LPF (L) dan High-Pass Filter atau HPF (H). Untuk ekstraksi fitur dalam sistem ini digunakan dekomposisi dengan varian *wavelet* Daubachies. Nilai L untuk Daubechies-4 (D4) yang akan digunakan disini adalah sebagai berikut:

$$\begin{aligned} h_0 &= (1 + \sqrt{3})/4\sqrt{2} \\ h_1 &= (3 + \sqrt{3})/4\sqrt{2} \\ h_2 &= (3 - \sqrt{3})/4\sqrt{2} \\ h_3 &= (1 - \sqrt{3})/4\sqrt{2} \end{aligned} \quad (2)$$

Sedangkan nilai H berhubungan dengan nilai L di atas yaitu sebagai berikut:

$$g_0 = h_3, g_1 = -h_2, g_2 = h_1, g_3 = -h_0 \quad (3)$$

Berdasarkan nilai L dan H di atas dibangun filter-filter DWT berdasarkan operasi matriks yang ditunjukkan di (4).

$$\begin{aligned} H_{ll} &= h^T h \\ H_{lh} &= h^T g \\ H_{hl} &= g^T h \\ H_{hh} &= g^T g \end{aligned} \quad (4)$$

Operasi DWT untuk suatu citra dilakukan berdasarkan algoritma berikut.

1. Nyatakan citra masukan sebagai matriks dua dimensi.
2. Konvolusikan matriks citra dengan filter yang diinginkan.
3. Lakukan *downsampling* matriks hasil dari tahap 2.

Algoritma DWT di atas dilakukan untuk setiap filter yang ada sehingga didapat 4 buah citra yang merupakan sub bidang dari citra awal. Subbidang-subbidang tersebut diberi label Low-Low (LL, didapat dari filter H_{ll}) yang berisi aproksimasi citra awal, Low-High (LH, didapat dari filter H_{lh}) yang berisi detail horizontal, High-Low (HL, didapat dari filter H_{hl}) yang berisi detail vertikal, dan High-High (HH, didapat dari filter H_{hh}) yang berisi detail diagonal. Sub bidang LL akan dijadikan *input* untuk proses dekomposisi berikutnya.

C. Rotated Wavelet Filter

Dalam proses ekstraksi tekstur, data karakteristik dari informasi diagonal memberikan tambahan signifikan dalam performa pengenalan fitur. *Rotated Wavelet Filter* bisa digunakan untuk merotasi *wavelet* filter awal. Hal ini dilakukan karena sub bidang HH di *wavelet* standar mengandung informasi diagonal tetapi susah dibedakan apakah informasi tersebut berdasarkan sudut 45° atau 135° . Apabila dilakukan rotasi sendiri maka bisa diatur arah sudut mana yang dihitung.

Filter yang digunakan dalam RWF didapatkan dengan cara merotasi filter yang bersesuaian dari filter-filter DWT sebesar sudut yang diinginkan. Dalam tugas akhir ini, sudut yang digunakan adalah 45° . Ukuran dari filter RWF adalah $(2N-1) \times (2N-1)$ dimana N adalah panjang dari filter 1-Dimensi [6]. Karena filter yang digunakan adalah D4 yang berarti panjang filter 1-D adalah 4 maka ukuran RWF adalah 7×7 .

Proses ekstraksi untuk RWF sama dengan yang dilakukan untuk DWT. Tetapi data yang digunakan hanya dari sub bidang LH dan HL karena hanya dua sub bidang ini yang mengandung informasi diagonal.

Fitur yang digunakan dari DWT dan RWF untuk merepresentasikan suatu citra adalah energi. Dasar pemikiran dari penggunaan energi sebagai fitur untuk pembedaan tekstur adalah bahwasanya distribusi energi dalam domain frekuensi mengidentifikasi sebuah tekstur [7]. Dekomposisi citra menggunakan DWT dan RWT dilakukan sampai 5 level. Energi dari DWT untuk masing-masing level, untuk setiap sub bidang (HH, HL, LH, LL) dihitung dan dijadikan vektor fitur dengan ukuran $5 \times 4 = 20$. Sedangkan dari RWT diambil Energi untuk masing-masing level dari sub bidang LH dan HL yang berarti ada $2 \times 5 = 10$ fitur. Persamaan untuk menghitung energi adalah sebagai berikut:

$$\text{Energi} = \frac{1}{M \times N} \sum_{i=1}^M \sum_{j=1}^N |X_{ij}| \quad (5)$$

D. Fuzzy C-Means Clustering

Setelah didapatkan fitur-fitur tekstur berdasarkan proses-proses di atas, maka akan dilakukan proses pengelompokan data (*Data Clustering*). *Data clustering* adalah proses membagi elemen-elemen data ke klaster-klaster sehingga elemen-elemen di klaster yang sama menjadi semirip mungkin, sedangkan elemen di klaster yang berbeda menjadi setidak mirip mungkin. Metode *clustering* yang akan digunakan adalah *Fuzzy C-Means* (FCM) yang telah dimodifikasi.

Algoritma FCM berusaha membagi sekumpulan data n ke dalam sejumlah c klaster. Hasil dari algoritma akan menghasilkan list pusat klaster $C = \{c_1, \dots, c_c\}$ dan matriks partisi $U = \mu_{ij}$, $i = 1, \dots, n$, $j = 1, \dots, c$, dimana setiap elemen μ_{ij} menunjukkan tingkat kedekatan elemen ke- i dengan klaster ke- j [8]. Persamaan untuk menghitung nilai μ_{ij} adalah sebagai berikut:

$$\mu_{ij} = \frac{1}{\sum_{k=1}^c \left(\frac{d_{ij}}{d_{ik}} \right)^{2/(m-1)}} \quad (6)$$

dimana

μ_{ij} : Nilai keanggotaan data ke-i dalam kluster ke-j

d_{ij} : Jarak data ke-i ke pusat kluster ke-j.

m : Parameter yang menunjukkan tingkat *fuzziness*, $m > 1$

c : Jumlah kluster

d_{ik} : Jarak data ke-i ke pusat kluster ke-k.

Algoritma FCM ini punya satu batasan utama, yaitu total keanggotaan suatu data ke semua kluster harus sama dengan satu.

Penghitungan pusat kluster c_j yang baru didasarkan pada nilai keanggotaan masing-masing data. Persamaannya adalah sebagai berikut.

$$c_j = \frac{\sum_{i=1}^n \mu_{ij}^m \cdot x_i}{\sum_{i=1}^n \mu_{ij}^m} \quad (7)$$

Secara umum algoritma FCM bisa dinyatakan sebagai berikut:

1. Tentukan nilai keanggotaan suatu data dengan semua kluster secara random
2. Ulangi langkah-langkah berikut ini sampai terjadi konvergensi.
 - a) Hitung pusat dari masing-masing kluster c_j berdasarkan (9).
 - b) Hitung keanggotaan setiap data untuk masing-masing kluster μ_{ij} berdasarkan (7).

Untuk menghitung jarak antara dua data digunakan metode *Canberra Distance*. Metode ini digunakan karena nilai-nilai dalam satu vektor fitur mempunyai skala yang berbeda-beda sehingga diperlukan normalisasi yang sudah diperhitungkan dalam *Canberra Distance*. Persamaannya adalah sebagai berikut.

$$d_{xy} = \sum_{a=1}^p \frac{|x_a - y_a|}{|x_a| + |y_a|} \quad (8)$$

Penghitungan evaluasi untuk kluster hasil dari sistem ini dilihat menggunakan beberapa metode, yaitu *Overall F-Measure* dan penghitungan kualitas pengklasteran.

F-measure adalah suatu tes akurasi yang menggunakan *precision* dan *recall* sebagai tolak ukurnya. *Precision* adalah nilai yang diperoleh dari pembagian hasil data yang benar dengan seluruh hasil data. Karena dalam sistem ini menggunakan *fuzzy clustering* dimana suatu data merupakan anggota lebih dari satu kluster, maka nilai *precision* akan kecil. Sedangkan *recall* diperoleh dari pembagian hasil data yang benar dibagi seluruh data yang seharusnya benar [9]. *F-measure* (F) dihitung dari *precision* p dan *recall* r berdasarkan (9).

$$F = 2 \cdot \frac{p \cdot r}{p + r} \quad (9)$$

Dalam sistem ini, untuk masing-masing kluster hasil pengklasteran dihitung nilai *F-Measure*-nya terhadap masing-

masing *dataset*. Selanjutnya dilakukan penghitungan *Overall F-Measure* ($F(C)$) sebagai nilai evaluasi sistem secara keseluruhan. Semakin tinggi nilai ini, maka semakin bagus hasil pengklasteran sistem tersebut [10]. $F(C)$ dihitung berdasarkan Persamaan (10).

$$F(C) = \sum_{K_i \in K} \frac{|K_i|}{|D|} \max_{c_j \in C} (F) \quad (10)$$

dimana

$|D|$: Total data dari *dataset*

C : Label kluster hasil dari sistem

K : Label kelas dari *dataset*

$|K_i|$: Jumlah data dalam kelas K_i

c_j : Label kluster ke-j hasil dari sistem

Sedangkan kualitas pengklasteran (biasa disebut validitas pengklasteran) dihitung berdasarkan rasio antara jarak intra kluster dan antar kluster. Jarak intra kluster diperoleh dengan cara menghitung rata-rata dari jumlah jarak masing-masing data dengan pusat klasternya. Sedangkan jarak antar kluster adalah jarak minimum antara masing-masing kluster. Semakin bagus hasil dari suatu pengklasteran, semakin kecil nilai validasinya.

III. HASIL DAN DISKUSI

Dataset yang digunakan berjumlah sebanyak 182 citra dengan jumlah motif 6. Untuk motif parang ada 29 citra, Ceplok 37 citra, Lereng 16 citra, Semen 13 citra, Lung-lungan 58 citra, dan Buketan 29 citra. *Dataset* ini diambil dari koleksi buku Danar Hadi karangan H. Santoso Doellah.

A. Uji Coba Pengklasteran

Hal pertama yang dilakukan dalam percobaan ini adalah menentukan kelas-kelas dari motif-motif dari *dataset* pengklasteran. Penentuan kelas dipilih berdasarkan persentase terbesar anggota kelompok asli dari *dataset* pengklasteran. Kemudian berdasarkan parameter yang diuji coba yaitu *fuzziness*, dilakukan penghitungan kualitas pengklasteran menggunakan *Overall F-Measure* dan validasi berdasarkan jarak intra kluster dan antar kluster.

Tabel 1.
Hasil Uji Coba Pengaruh *Fuzziness*

<i>Fuzziness</i>	<i>Overall F-Measure</i>	Validasi
1.1	0,315	0,007375
1.5	0,348	0,020843
2	0,364	0,005208

Rangkuman data hasil percobaan bisa dilihat pada Tabel 1. Penggunaan *fuzziness* yang berbeda memberikan pengaruh yang cukup signifikan. Nilai yang mendekati satu (yang berarti mirip dengan *k-means*) memberikan *Overall F-Measure* yang paling rendah. Hal ini bisa dimaklumi karena dengan *fuzziness* kecil, maka data-data semakin terikat sehingga jika di awal inisialisasi ada data yang salah masuk kluster, maka sampai selesai proses *clustering* data tersebut tidak akan terpisah dari kluster itu meskipun sebenarnya salah. Peningkatan nilai *fuzziness* memberikan kenaikan *Overall F-*

Measure dan hasil validasi yang lebih baik.

B. Analisa untuk Batik Multi-Label

Percobaan hanya dilakukan berdasarkan pengamatan visual saja, sehingga ada kemungkinan hasilnya tidak objektif. Data yang digunakan dalam percobaan ini hanya beberapa buah data. Sebagai contoh akan dilakukan percobaan untuk citra batik pada Gambar 1.



Gambar 1. Batik Patran Kangkung

Dari perhitungan didapatkan citra ini mempunyai label-label Lereng, Parang, dan Buketan. Citra ini mempunyai unsur lereng dengan motif diagonalnya, parang karena selain diagonal salah satu motif daun dianggap motif mlinjon, sedangkan detail daun-daun yang membentuk motif diagonal merupakan motif buketan.

Percobaan ini dilakukan beberapa kali pada citra-citra yang berbeda untuk mendapatkan gambaran mengenai kualitas penggunaan *fuzzy c-means* dalam pengenalan motif *multi-label*. Dari percobaan-percobaan tersebut diambil kesimpulan bahwa metode *fuzzy c-means* cukup bisa digunakan untuk mengenali batik multi label.

IV. KESIMPULAN

Kesimpulan yang bisa diambil dari tugas akhir ini adalah sebagai berikut.

- 1) *Fuzzy C-Means Clustering* tidak terlalu cocok digunakan dalam pengklasteran data yang terlalu tersebar seperti motif batik ini. Di sisi lain penggunaan FCM untuk pengenalan batik *multi-label* cukup memuaskan.
- 2) Penentuan nilai *fuzziness* mempengaruhi nilai *Overall F-Measure*. *Overall F-Measure* tertinggi didapatkan saat penggunaan *fuzziness* sebesar 2 dengan nilai 0,364.

DAFTAR PUSTAKA

- [1] B. Arisandi, N. Suciati, dan A.Y. Wijaya. *Pengenalan Motif Batik dengan Rotated Wavelet Filter dan Neural Network*. Jurusan Teknik Informatika, ITS, Indonesia. 2011.
- [2] Santosa Doellah. *Batik : Pengaruh Zaman dan Lingkungan*. Batik Danar Hadi Solo. 2002.
- [3] G. Hoffmann. *Luminance Models for Grayscale Conversions*. 2008.
- [4] D. Hazra. "Texture recognition with combined GLCM, wavelet and rotated wavelet features". *International Journal of Computer and Electrical Engineering*, Vol.3, No.1, February, 2011, pp. 1793-8163.
- [5] I. Daubechies. *Ten Lectures on Wavelets*. Philadelphia: PA. 1992.
- [6] M. Kokare, P. Biswas, dan B. Chatterji. "Texture image retrieval using rotated wavelet filter". *Pattern Recognition Letters* 28, 2007, pp. 1240–1249.
- [7] M. Kokare, P. Biswas, dan B. Chatterji. *Wavelet Transform Based Texture Features For Content Based Image Retrieval*. Kharagpur: Electronics and Electrical Communication Engineering Department, Indian Institut of Technology. 2003.
- [8] B. Thomas, G. Raju, dan S. Wangmo. "A modified fuzzy c-means algorithm for natural data exploration". *World Academy of Science, Engineering and Technology* 49, 2009.
- [9] M.E.S. Mendes, dan L. Sack. "Evaluating fuzzy clustering for relevance-based information". *IEEE International Conference on Fuzzy Systems*, Mei, 2003, pp. 648 – 653.
- [10] B.C. Fung. *Hierarchical Document Clustering Using Frequent Itemsets*. Master Thesis, Simon Fraser University, Computing Science. 2002.