

## UNJUK KERJA K-NEAREST NEIGHBORS PADA PENGENALAN KARAKTER JAWA BERBASIS LOCAL BINARY PATTERN

Ajib Susanto<sup>1\*</sup>, Daurat Sinaga<sup>2</sup>, Eko Hari Rachmawanto<sup>3</sup>, De Rosal Ignatius Moses Setiadi<sup>4</sup>

<sup>1,2,3,4</sup>Jurusan Teknik Informatika, Fakultas Ilmu Komputer, Universitas Dian Nuswantoro

Jl. Imam Bonjol 207, Kode Pos 50131, Semarang, Jawa Tengah

\*Email: ajib.susanto@dsn.dinus.ac.id

### Abstrak

Aksara jawa merupakan salah satu warisan budaya di Indonesia yang umumnya berasal dari Jawa Tengah dan Yogyakarta. Pemahaman akan aksara jawa terutama pada siswa di sekolah dirasa masih kurang karena sulitnya membedakan bentuk aksara jawa dengan huruf alfabet. Maka dari itu pada makalah ini mengusulkan metode pengenalan aksara jawa menggunakan Local Binary Pattern (LBP) dan K-Nearest Neighbors (KNN). LBP digunakan untuk mengekstrak ciri unik dari citra tulisan aksara jawa. Sementara KNN digunakan untuk menentukan kelas dari citra aksara jawa berdasarkan hasil ekstraksi ciri LBP. Pengujian dilakukan dengan menggunakan 160 dataset citra yang dibagi menjadi 40 citra uji dan 120 citra latih. Evaluasi hasil pengujian menggunakan perhitungan confusion matrix untuk menentukan akurasi dari kombinasi tersebut. Dari hasil pengujian didapatkan akurasi tertinggi adalah 82,5% dimana parameter yang digunakan adalah cell size berukuran 64x64 dan nilai  $k = 3$ .

**Kata kunci:** aksara jawa, KNN, LBP

### 1. PENDAHULUAN

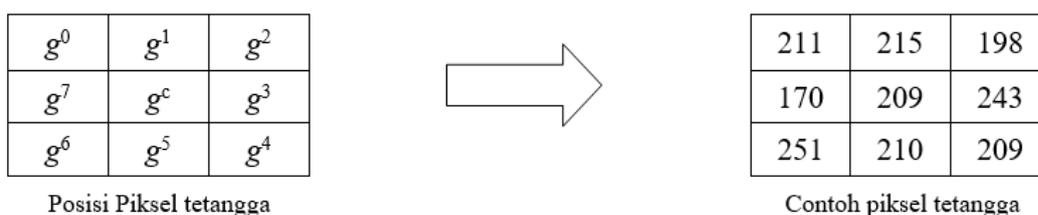
Bahasa jawa merupakan salah satu bahasa endemik yang berasal dari Jawa Tengah dan Yogyakarta. Sekarang ini, penulisan bahasa jawa ditulis ke dalam bentuk huruf latin atau alfabetik. Namun sebenarnya bahasa jawa memiliki huruf atau aksara sendiri yang disebut dengan aksara jawa *nglegena / hanacaraka* (Sebastian, Budhi dan Adipranata, 2015). Aksara jawa memiliki bentuk huruf yang sangat berbeda dengan huruf alfabetik/ latin. Penggunaan aksara jawa sudah jarang diimplementasikan ke dalam dokumen tertulis. Pengetahuan tentang aksara jawa kini hanya diajarkan sebagai materi muatan lokal di sekolah-sekolah di daerah Jawa Tengah dan Yogyakarta. Oleh karena perbedaan bentuk huruf dan hanya dipelajari sebagai materi tambahan di sekolah menyebabkan pengenalan siswa terhadap aksara jawa sangat rendah (Supriyono et al., 2016).

Dalam implementasinya, teknologi seperti komputer dapat menjadi salah satu pilihan yang dapat digunakan untuk membantu siswa mengenali aksara jawa dengan mudah. Komputer akan mengenali pola aksara jawa berdasarkan citra masukkan dan akan menampilkan keluaran berupa terjemahan huruf latin dari aksara tersebut. Proses pengenalan pola pada komputer membutuhkan proses pengambilan ciri pada pola citra aksara jawa. Salah satu jenis algoritma pengambilan ciri adalah *Local Binary Pattern* (LBP). LBP akan diimplementasikan pada citra untuk mendapatkan ciri unik dari pola citra tersebut. LBP sendiri merupakan metode yang membandingkan pusat blok citra dengan piksel tetangganya. Sari dkk (Sari, Hidayat dan Sunarya, 2015) mengimplementasikan LBP serta membandingkan algoritma *Support Vector Machine* (SVM) dan *K-Nearest Neighbor* (KNN) dalam proses pengenalan aksara bali. Dalam proses pengenalan pola, SVM dan KNN digunakan sebagai penentu *class* tiap pola citra yang sebelumnya telah diekstraksi menggunakan LBP. Berdasarkan penelitian tersebut, pada makalah ini penulis mengajukan metode pengenalan aksara jawa menggunakan ekstraksi fitur *Local Binary Pattern* (LBP) dan algoritma klasifikasi *K-nearest Neighbor* (KNN).

## 2. LANDASAN TEORI

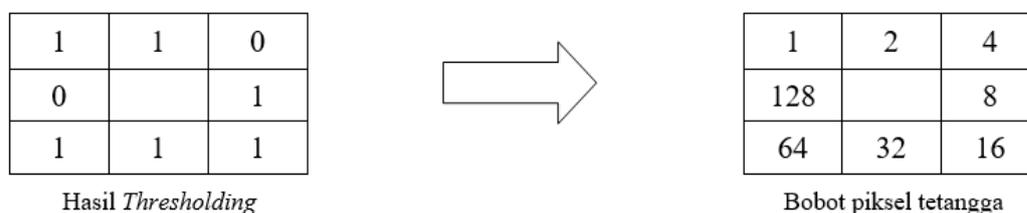
### 2.1. Local Binary Pattern (LBP)

LBP atau *Local Binary Pattern* merupakan salah satu algoritma *pre-processing* yang digunakan dalam proses pengenalan pola. Algoritma ini akan mengekstrak informasi ciri dari suatu citra yang kemudian hasil ekstraksinya akan dipetakan menggunakan algoritma *data mining*. LBP akan membandingkan pusat citra dengan 8 piksel tetangganya ( $g^n$ ) yang dibagi menjadi  $g^0$  hingga  $g^7$  seperti yang ditunjukkan oleh Gambar 1 (Hayaty *et al.*, 2017). Untuk melakukan ekstraksi LBP langkah pertama adalah jika citra masukkan masih berupa citra berwarna maka harus diubah menjadi citra *grayscale*. Tentukan pusat citra lalu bandingkan dengan 8 piksel tetangganya. Penentuan piksel tetangga dihitung searah jarum jam sesuai Gambar 1.



**Gambar 1. Penentuan nilai piksel ketetanggaan**

Setelah nilai piksel tetangganya didapat, kemudian lakukan *threshold* pada citra tersebut dengan nilai batas *threshold*-nya sama dengan nilai  $g^c$ . Lalu beri bobot pada masing-masing tetangga ( $g^n$ ) dengan pembobotan sesuai dengan nilai  $2^n$  sesuai Gambar 2.



**Gambar 2. Penghitungan LBP pada citra berdasarkan hasil *thresholding*-nya**

Nilai LBP ditentukan dengan menggunakan hasil *thresholding* dan tiap bobot tetangganya yang ditunjukkan pada Gambar 2. Jika nilai *threshold* nya bernilai 1 maka bobot yang didapat adalah maksimal, sementara jika bernilai 0 maka bobotnya juga bernilai 0.

$$\text{Hasil threshold} = 1\ 1\ 0\ 1\ 1\ 1\ 1\ 0$$

$$\text{LBP} = 1 + 2 + 0 + 8 + 16 + 32 + 64 + 0 = 123$$

Maka dari hasil perhitungan LBP di atas didapatkan bahwa nilai LBP pada blok citra tersebut adalah 123.

### 2.2. K-Nearest Neighbor (KNN)

KNN merupakan salah satu algoritma *supervise learning* yang digunakan dalam proses klasifikasi. Algoritma ini akan membandingkan *testing set* dengan *training set* untuk mendapatkan informasi mengenai jarak terdekat (Leidiyana, 2013). Setelah itu, untuk menentukan kelas nya digunakan variabel  $k$  dimana nilai  $k$  merupakan jumlah perbandingan kelas dengan jarak terdekat. Umumnya nilai  $k$  merupakan bilangan ganjil (1,3,5,7, dst). Secara sederhana KNN dibagi menjadi 2 bagian utama yakni proses pencarian jarak terdekat antara data uji (*testing set*) dan data latih (*training set*) serta penentuan kelas berdasarkan nilai  $k$  yang digunakan.

Secara umum, pada saat menentukan jarak antara data uji dengan data latih pada KNN digunakan perhitungan jarak *euclidean*. Berikut adalah persamaan yang digunakan untuk menghitung jarak *euclidean*:

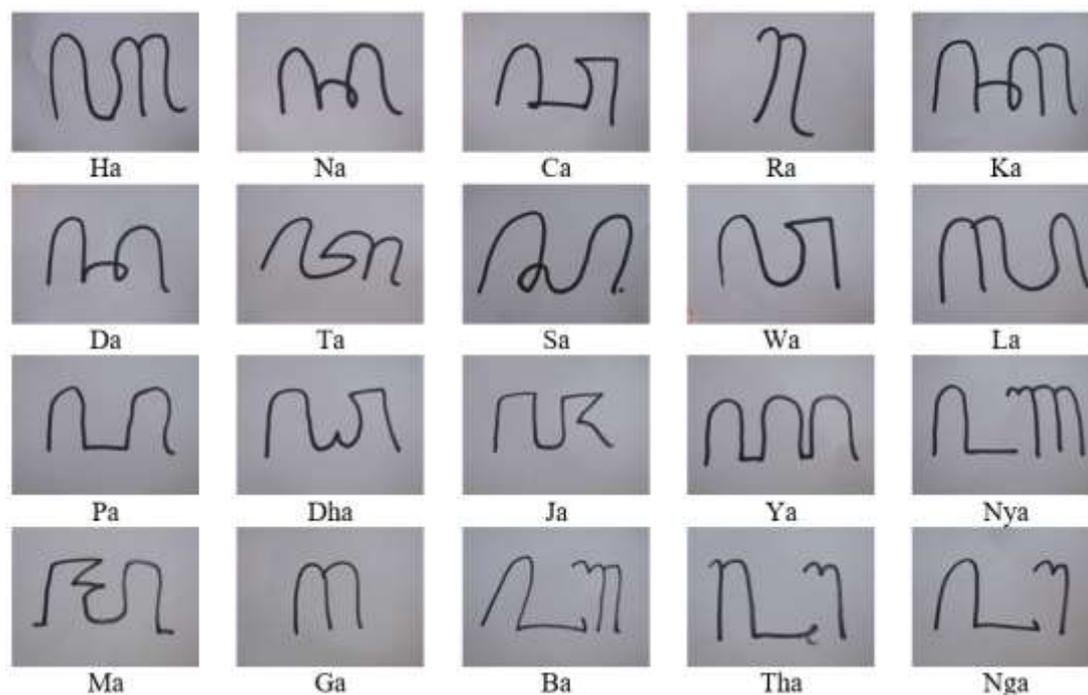
$$E_{(i,j)} = \sqrt{\sum_{d=1}^n (i_d - j_d)^2} \quad (1)$$

Pada persamaan 1 diketahui bahwa  $E_{(i,j)}$  merupakan jarak *euclidean* antara data uji  $i_d$  dengan data latih  $j_d$  dimana  $d$  merupakan banyak atribut yang akan dihitung jaraknya (Indriani *et al.*, 2017).

### 3. METODOLOGI

#### 3.1. Dataset Citra Aksara Jawa

Pada makalah ini citra yang akan digunakan adalah citra aksara jawa dimana aksara jawa tersebut berupa tulisan tangan sebanyak 160 citra. Berikut merupakan 20 contoh dari citra uji yang digunakan:



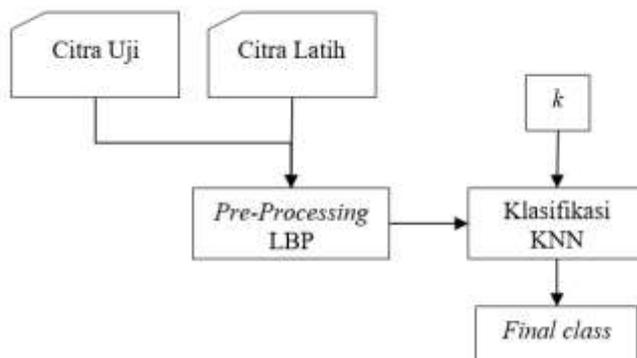
**Gambar 3. Contoh citra uji yang digunakan**

Pada Gambar 3 ditunjukkan 20 contoh citra tulisan aksara jawa yang akan digunakan dalam penelitian ini. Dari 180 citra yang digunakan, 40 diantaranya merupakan citra uji, sementara 120 sisanya merupakan citra latih.

#### 3.2. Metode yang Diajukan

Pada penelitian ini LBP akan dikombinasikan dengan KNN untuk mengklasifikasikan citra aksara jawa untuk jenis aksara *hanacaraka* atau *nlegena*. Kombinasi antara LBP dan KNN akan disimulasikan menggunakan program *Matlab 2015b*. Langkah pertama dalam proses klasifikasinya, LBP akan diimplementasikan pada citra uji maupun citra latih guna menghasilkan ciri tiap citra. Selanjutnya tentukan nilai  $k$ . Kemudian, berdasarkan ciri tadi hitung jarak *euclidean* menggunakan persamaan 1. Setelah didapatkan jarak antara data uji

dan data latih, tentukan  $k$  jarak terdekat lalu tentukan kelas yang memiliki tingkat kemunculan paling banyak. Kelas yang muncul paling banyak merupakan kelas hasil klasifikasi menggunakan KNN. Secara sederhana, proses pada metode yang diajukan dapat dilihat pada gambar 4.



**Gambar 4. Flowchart metode yang diajukan**

Untuk mengetahui performa dari kombinasi LBP dan KNN dilakukan pengujian menggunakan *confusion matrix*. *Confusion matrix* merupakan salah satu pengujian yang memberikan performa akurasi dari suatu algoritma klasifikasi. Berikut proses perhitungan akurasi menggunakan *confusion matrix* (Sutojo *et al.*, 2017) sesuai Tabel 1.

**Tabel 1. Penentuan tabel *confusion matrix* berdasarkan hasil klasifikasi**

		<i>Predict result</i>	
		Yes	No
<i>Actual result</i>	Yes	TP	FN
	No	FP	TN

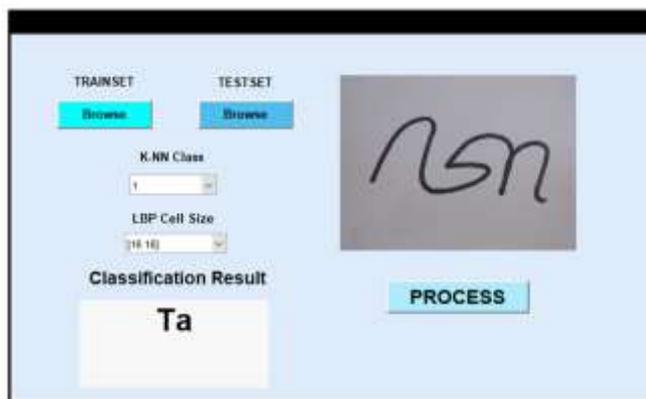
Pada Tabel 1 variasi kelas yang digunakan hanya 2 sehingga matriks yang digunakan hanya 2x2 saja. Jika terdapat 5 variasi kelas, maka matriks yang digunakan berukuran 5x5. Pada Tabel 1 ditunjukkan bahwa untuk menghitung akurasi menggunakan *confusion matrix* perlu adanya *predict result* dan *actual result*. *Predict result* merupakan kelas hasil klasifikasi menggunakan LBP dan KNN, sementara *actual result* merupakan kelas asli dari citra uji. Selanjutnya, untuk menghitung akurasi digunakan rumus berikut:

$$Accuracy = \frac{TP + TN}{FP + TP + TN + FN} \tag{2}$$

Persamaan 2 menunjukkan rumus perhitungan akurasi berdasarkan tabel *confusion matrix* pada Tabel 1.

**4. HASIL DAN PEMBAHASAN**

Hasil pengujian dilakukan dengan beberapa kondisi parameter, yakni ukuran *cell* (16x16, 32x32, 64x64, 128x128) pada LBP dengan jumlah tetangga yang berpengaruh sebesar 8, serta parameter  $k$  pada KNN yang diuji cobakan adalah 1, 3, 5, dan 7. Berikut merupakan contoh simulasi tampilan program *Matlab* 2015b pada saat mengklasifikasikan citra aksara ‘Ta’ dengan menggunakan LBP dan KNN:



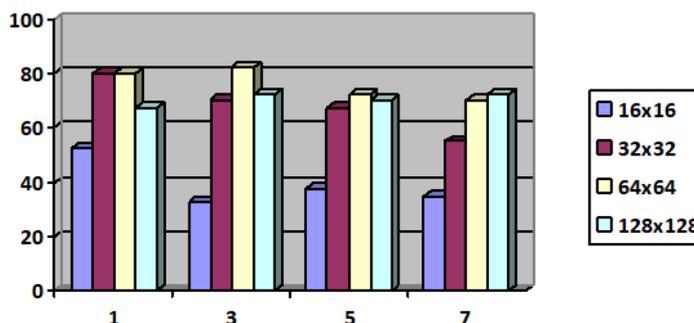
Gambar 5. Simulasi klasifikasi citra aksara ‘Ta’

Dapat dilihat pada Gambar 5, citra ‘Ta’ diklasifikasikan menggunakan parameter *cell size* dengan ukuran 16×16 dan *k* sama dengan 1 dimana jumlah tetangga yang berpengaruh adalah 8. Perhitungan akurasi menggunakan *confusion matrix* memerlukan tabel berukuran 20×20 karena jumlah kelas yang digunakan sebanyak 20 kelas.



Gambar 6. Simulasi perhitungan akurasi

Pada Gambar 6 ditunjukkan tampilan simulasi hasil perhitungan akurasi yang didapatkan adalah 80% dan parameter *cell size* berukuran 32×32 dan *k* = 1. Selain itu perhitungan akurasi menggunakan *confusion matrix* dengan 40 data uji serta 120 data latih dilakukan dengan beberapa skema parameter yang berbeda untuk mengetahui kombinasi dengan tingkat akurasi terbaik. Berikut contoh Tabel pengujian dengan skema parameter yang berbeda ditunjukkan oleh Tabel 2 dan Gambar 7.



Gambar 7. Diagram perbandingan hasil akurasi

**Tabel 2. Hasil pengujian dengan skema parameter berbeda**

<i>Cell Size</i>	<i>k</i>	Akurasi (%)
16×16	1	52,5
	3	32,5
	5	37,5
	7	35
32×32	1	80
	3	70
	5	67,5
	7	55
64×64	1	80
	3	82,5
	5	72,5
	7	70
128×128	1	67,5
	3	72,5
	5	70
	7	72,5

Tabel 2 menunjukkan skema pengujian dengan menggunakan kombinasi parameter LBP dan KNN yang berbeda telah dilakukan. Dapat diketahui bahwa pada tabel tersebut akurasi tertinggi sebesar 82,5% didapat dengan menggunakan kombinasi *cell size* dengan ukuran 64×64 dan  $k = 3$ .

## 5. KESIMPULAN

Pada penelitian ini telah dilakukan kombinasi LBP dengan KNN untuk mengklasifikasikan aksara jawa *nglegena* atau *hanacaraka* berdasarkan citra aksaranya. Citra aksara yang digunakan merupakan citra aksara yang ditulis dengan tangan yang berjumlah 160 citra dimana terdiri dari 40 citra uji dan 120 citra latih. Hasil pengujian menunjukkan kombinasi parameter *cell size* berukuran 64×64 dan parameter  $k = 3$  memberikan akurasi tertinggi sebesar 82,5%. Penggunaan LBP pada proses pengenalan aksara jawa dapat menjadi pilihan metode pengenalan pola yang kedepannya dapat diimplementasikan untuk meningkatkan pemahaman siswa sekolah terhadap aksara jawa.

## DAFTAR PUSTAKA

- Hayaty, N. *et al.*, (2017), "Jurnal Sustainable : Jurnal Hasil Penelitian dan Industri Terapan Penerapan Algoritma Local Binary Pattern untuk Pengenalan Pola Sidik Jari," *Jurnal Sustainable: Jurnal Hasil Penelitian dan Industri Terapan*, 6(2), hal. 6–11.
- Indriani, O. R. *et al.*, (2017), "Tomatoes classification using K-NN based on GLCM and HSV color space," in *2017 International Conference on Innovative and Creative Information Technology (ICITech)*, hal. 1–6. doi: 10.1109/INNOCIT.2017.8319133.
- Leidiyana, H., (2013), "Penerapan Algoritma K-Nearest Neighbor Untuk Penentuan Resiko Kredit Kepemilikan Kendaraan Bermotor," *Jurnal Penelitian Ilmu Komputer, System Embedded & Logic*, 1(1), hal. 65–76.
- Sari, I. A. D. P., Hidayat, B., dan Sunarya, U., (2015), "Pengenalan Aksara Bali dengan Metode Local Binary Pattern," *E-Proceeding of Engineering*, 2(2), hal. 2697–2704.
- Sebastian, B., Budhi, G. S., dan Adipranata, R., (2015), "Penggunaan Decision Tree Dengan ID3 Algorithm Untuk Mengenali Dokumen Beraksara Jawa," *Jurnal Infra*, 3, hal. 142.
- Supriyono, H. *et al.*, (2016), "Rancang Bangun Media Pembelajaran Dan Game Edukatif Pengenalan Aksara Jawa ‘ Pandawa ,’" *Prosiding The 4thUniversity Research Colloquium 2016*, hal. 1–12.
- Sutojo, T. *et al.*, (2017), "CBIR for classification of cow types using GLCM and color features extraction," *2017 2nd International conferences on Information Technology, Information Systems and Electrical Engineering (ICITISEE)*, hal. 182–187. doi: 10.1109/ICITISEE.2017.8285491.