PENGENALAN AKSARA BALI DENGAN METODE LOCAL BINARY PATTERN

BALI SCRIPT RECOGNITION WITH LOCAL BINARY PATTERN METHOD

Ida Ayu Dian Purnama Sari[1] Dr. Ir. Bambang Hidayat, DEA[2] Unang Sunarya, S.T., M.T[3]

[1],[2],[3]Fakultas Teknik Elektro – Universitas Telkom
Jl. Telekomunikasi, Dayeuh Kolot Bandung 40257 Indonesia
[1]idaayudianpurnamasari@gmail.com, [2] bhidayat@telkomuniversity.ac.id
[3] unangsunarya@telkomuniversity.ac.id

Abstrak

Aksara Bali merupakan salah satu tulisan daerah yang digunakan di Indonesia. Untuk membaca aksara Bali tidaklah mudah. Hal ini disebabkan karena kesulitan untuk mengenali bentuk atau pola dari suku kata dasar aksara tersebut. Suku kata dasar aksara Bali hampir memiliki struktur yang sama, sehingga di dalam pembacaannya akan menemukan kesulitan mengidentifikasi suku kata dasar. Penelitian ini dibuat dengan tujuan untuk membuat suatu sistem yang mampu mengenali pola dari aksara Bali dimana aksara Bali yang digunakan adalah aksara Bali dasar. Sistem ini menggunakan teknik *Local Binary Pattern* (LBP) untuk ekstraksi cirinya. Salah satu sifat paling penting dari operator LBP ialah kesederhanaan perhitungannya, memiliki waktu komputasi yang lebih cepat, dan sifatnya yang invarian terhadap perubahan fotometri dari objek yang sama, dikarenakan LBP merupakan ukuran intensitas relatif suatu piksel dengan intensitas piksel disekitarnya. LBP didefinisikan sebagai perbandingan nilai piksel pada pusat citra dengan nilai piksel disekelilingnya sehingga diperoleh nilai biner pada matriks tersebut. Hasil dari LBP ini dijadikan masukan pada proses klasifikasi citra yang menggunakan *Support Vector Machine* dan *K-Nearest Neighbor*. Keunggulan dari metode KNN adalah tangguh terhadap data latih yang memiliki banyak *noise* dan efektif apabila data latihnya berukuran besar, sedangkan metode klasifikasi SVM mempunyai beberapa kelebihan, diantaranya bisa memodelkan dan mengklasifikasikan hubungan antar variabel tanpa perlu asumsi yang ketat, efisien, dan interpretasinya mudah. Pada penelitian ini juga membandingkan proses klasifikasi antara metode *Support Vector Machine* dengan *K- Nearest Neighbor*. Hasil dari simulasi yang dilakukan sistem dapat mengenali Aksara Bali dengan tingkat akurasi tertinggi adalah 74,6%, ddengan waktu komputasi rata-rata sistem sebesar 2,3203 detik.

Kata kunci : Aksara Bali, *Local Binary Pattern* (LBP), *Support Vector Machine* (SVM), *K-Nearest Neighbor* (KNN).

Abstract

Balinese script is one of the texts used in the Indonesian region. To read Balinese script is not easy. It is due to the difficulty to recognize the shape or pattern of syllables from the basic script. Syllable basic from the Balinese script almost has the same structure, so that in its reading will find the difficult to identify syllables basic. The aims of study is to implementation a system that is able to recognize the texture from the Balinese Script where is using to be a basic of Balinese script. This system will using a Local Binary Pattern (LBP) technique for the characteristics extraction. One of the most important properties of LBP operator is the simplicity of calculation, has a faster computation time, and it is invariant to changes in the photometry of the same object, due to LBP is a measure of the relative intensity of a pixel by pixel intensities around. LBP is defined as the ratio of the value of the central pixel in the image with the surrounding pixel values in order to obtain a binary value in the matrix. The results of this LBP will be used as inputs in the process of image classification using Support Vector Machine (SVM) and K-Nearest Neighbor (KNN). The advantages of the method KNN is resilient against training data that has a lot of noise and effective if the data latihnya large, while the SVM classification method has several advantages, including design and can classify the relationship between variables and without the assumption of a strict, efficient, and easy interpretation. In this research, also compare the process of classification between Support Vector Machine with K-Nearest Neighbor method. The results of the

simulation system from Balinese script recognition with highest accuracy system 74,6 %, with the average computation time for 2,3203 seconds.

Keywords : *Balinese Script, Local Binary Pattern (LBP), Support Vector Machine (SVM), and K-Nearest Neighbor (KNN).*

1. Pendahuluan

Aksara Bali adalah salah satu kekayaan budaya Indonesia yang harus dilestarikan. Aksara Bali sangat erat kaitannya dengan kehidupan budaya dan agama Hindu di Bali. Seiring dengan perkembangan waktu, aksara Bali mulai kurang diminati oleh generasi muda. Hal ini disebabkan karena kesulitan untuk mengenali bentuk atau pola dari suku kata dasar aksara tersebut. Suku kata dasar aksara Bali hampir memiliki struktur yang sama, sehingga di dalam pembacaannya akan menemukan kesulitan mengidentifikasi suku kata dasar[13].

Pada penelitian ini, dikembangkan suatu sistem dari penelitian sebelumnya yang berjudul *Identifikasi Aksara Bali ke Huruf Latin dengan Menggunakan Klasifikasi Template Matching dan K-Nearest Neighbor*, dimana pada penelitian sebelumnya menggunakan data latih berupa kata dalam aksara Bali yang dilansir dari komputer. Metode yang digunakan untuk merancang sistem pengenalan aksara Bali, pada penelitian ini yaitu *Local Binary Pattern* untuk ekstraksi ciri dan menggunakan metode klasifikasi *Support Vector Machine* dan *K-Nearest Neighbor*. *Local Binary Pattern* merupakan metode yang membandingkan piksel tetangga dengan piksel pusat sehingga diperoleh nilai biner pada matriks, lalu nilai biner tersebut dihitung dan diubah kedalam bentuk desimal. Salah satu sifat paling penting dari operator LBP ialah kesederhanaan perhitungannya, memiliki waktu komputasi yang lebih cepat, dan sifatnya yang invarian terhadap perubahan fotometri dari objek yang sama, dikarenakan LBP merupakan ukuran intensitas relatif suatu piksel dengan intensitas piksel disekitarnya. Keunggulan dari metode KNN adalah KNN tangguh terhadap data latih yang memiliki banyak *noise* dan efektif apabila data latihnya berukuran besar, sedangkan SVM merupakan salah satu metode klasifikasi yang mempunyai beberapa kelebihan, diantaranya bisa memodelkan dan mengklasifikasikan hubungan antar variabel tanpa perlu asumsi yang ketat, efisien, dan interpretasinya mudah. Pada penelitian ini, juga membandingkan proses klasifikasi ciri antara metode *Support Vector Machine* dengan *K-Nearest Neighbor*.

Sistem ini membandingkan hasil citra input dengan data yang sudah ada di dalam database. Output sistem berupa kata dan kalimat dalam tulisan huruf latin. Diharapkan metode yang digunakan pada penelitian ini dapat dijadikan sebagai salah satu metode pengenalan karakter yang handal.

2. Dasar Teori

2.1 Pengertian Aksara Bali

Aksara Bali adalah salah satu huruf tradisional Nusantara yang berkembang di Bali. Aksara ini hampir mirip dengan aksara Jawa. Perbedaannya terletak pada lekukan bentuk huruf. Aksara Bali berjumlah 47 karakter, 14 di antaranya merupakan aksara suara atau vokal. Aksara wianjana atau huruf konsonan berjumlah 33 karakter. Aksara wianjana Bali yang biasa digunakan berjumlah 18 karakter[8].

Gambar 1 Aksara wianjana [1]

2.2 Aksara Latin

Huruf Latin adalah alfabet yang pertama kalinya digunakan oleh orang Romawi untuk menuliskan bahasa Latin kira-kira sejak abad ke-7 sebelum Masehi. Pada saat ini alfabet Latin adalah aksara yang paling banyak dipakai di dunia untuk menuliskan berbagai bahasa, termasuk bahasa Indonesia. Beberapa negara mengadopsi dan memodifikasi alfabet Latin sesuai dengan fonologi bahasa mereka, karena tidak semua fonem dapat dilambangkan dengan huruf Latin. Alfabet Latin hanya mengandung 26 huruf, yaitu [8]:

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

2.3 LBP (*Local Binary Pattern*)

Operasi LBP dasar dengan asumsi bahwa tekstur memiliki 2 aspek yang berkomplemen, yaitu polanya dan dayanya. LBP (*Local Binary Pattern*) merupakan suatu operasi image yang mentransformasikan citra menjadi sebuah susunan label integer yang menggambarkan kenampakan skala kecil dari citra. Kemudian digunakan lagi untuk analisis citra yang lebih lanjut. LBP diusulkan sebagai unit tekstur versi dua tingkat untuk menggambarkan pola tekstur lokal[2][6].

Versi asli dari operasi LBP bekerja pada blok piksel 3x3 dari sebuah citra. Piksel-piksel di blok tersebut kemudian diberikan *threshold* oleh piksel tengah, lalu dikalikan kuadrat dua, dan kemudian dijumlahkan untuk mendapatkan label baru untuk piksel tengah. Karena sebuah ketetanggaan dari piksel terdiri dari delapan piksel, sejumlah 2^8=256 label yang berbeda yang mungkin didapatkan bergantung kepada nilai keabuan relatif dari piksel tengah pada ketetanggaan piksel[2].

2.3.1 Penurunan dari Operasi LBP Umum[2]

Secara sederhana, LBP adalah sebuah kode biner yang menggambarkan pola tekstur citra. Hal ini dibangun dengan lingkungan batas dengan nilai abu-abu dari pusatnya. Contoh komputasi LBP pada 3x3 pixel:

9	3	1
10	7	5
8	3	7

threshold

1	0	0
1		0
1	0	1

bobot

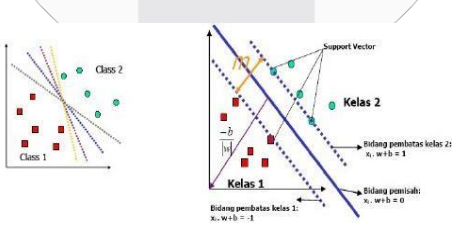
1	2	4
128		8
64	32	16

Pola = 11110001
LBP = 1+32 +64 +128 =225

Setiap pixel memiliki nilai hasil grayscale, kemudian dilakukan threshold berpusat pada titik tengah. Pixel yang memiliki nilai sama atau lebih dibandingkan dengan titik tengah diberi nilai 1 selain itu diberi nilai 0.Kemudian nilai LBP didapat dari penjumlahan dua pangkat nilai angka yang bernilai 1.

2.4 SVM (*Support Vector Machine*)

SVM adalah salah satu teknik klasifikasi data dengan proses pelatihan (*supervised learning*). Salah satu ciri dari metode klasifikasi SVM adalah menemukan garis pemisah (*hyperplane*) terbaik sehingga diperoleh ukuran margin yang maksimal. Margin adalah jarak antara *hyperplane* tersebut dengan titik terdekat dari masing-masing kelas atau dua kali jarak antara *hyperplane* dengan *support vector*. Titik yang terdekat dengan *hyperplane* disebut *support vector*[4].



Gambar 2 *Alternative* bidang pemisah (kiri) dan *alternative* bidang pemisah terbaik [5]

Adapun data yang berada pada bidang pembatas ini disebut *support vector*. Dalam contoh di atas, dua kelas dapat dipisahkan oleh sepasang bidang pembatas yang sejajar. Bidang pembatas pertama membatasi kelas pertama sedangkan bidang pembatas kedua membatasi kelas kedua, sehingga diperoleh:

$x_i \cdot w+b \geq +1$ *for* $y_i = +1$
$x_i \cdot w+b \leq -1$ *for* $yi = -1$ (1)

w adalah normal bidang dan b adalah posisi bidang relatif terhadap pusat koordinat. Nilai *margin* antara bidang pembatas adalah[4] :

———————— = —— (2)

Untuk mengklasifikasikan data yang tidak dapat dipisahkan secara linier formula SVM harus dimodifikasi karena tidak akan ada solusi yang ditemukan. Sulit mengetahui fungsi transformasi yang tepat. Untuk mengatasi masalah ini, pada SVM digunakan "*kernel trick*". Fungsi kernel yang umum digunakan adalah sebagai berikut[3]:

a. Kernel Linier
   Kernel Linier digunakan ketika data yang akan diklasifikasi dapat terpisah dengan sebuah hyperplane.
b. *Polynomial kernel*
   Kernel Linier digunakan ketika data yang akan diklasifikasi tidak dapat terpisah dengan sebuah garis/hyperplane (*non linier*). Kernel polynomial memiliki orde lebih dari satu mengakibatkan *support vector* yang diperoleh memiliki jarak yang lebih jauh antar kelasnya sehingga lebih mudah untuk menemukan *hyperplane* yang terbaik.
c. *Radial Basis Function* (RBF)
   Kernel Linier digunakan ketika data yang akan diklasifikasi tidak dapat terpisah dengan sebuah garis/hyperplane. Kernel RBF hanya memiliki satu orde sehingga *support vector* yang diperoleh lebih sedikit dan mengakibatkan sulit untuk menemukan *hyperplane* yang terbaik.

2.5 *K-Nearest Neighbor* (KNN)

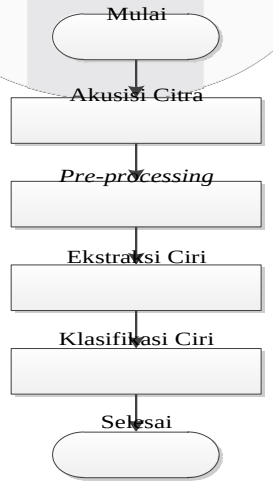
KNN adalah metode yang digunakan untuk mengklasifikasikan suatu data. Pada data latih biasanya diambil lebih dari satu tetangga terdekat dengan data uji kemudian digunakan algoritma ini untuk ditentukan kelasnya. Jadi klasifikasi *k-NN* mempunyai dua langkah, yaitu menentukan jarak tetangga terdekat dari data tersebut lalu menentukan kelas dari masing - masing tetangga terdekat tersebut. Pada penelitian ini digunakan jenis jarak KNN euclidean distance, yaitu[7]:

Euclidean Distance :

$$\sqrt{\sum} \quad (3)$$

3. Pembahasan

3.1 Perancangan Sistem



Gambar 3 Diagram Alir Umum Sistem Pengenalan Aksara Bali

Proses pertama dilakukan pengambilan citra aksara Bali. Setelah itu dilakukan *pre-processing* dimana citra akan melewati beberapa proses, sehingga citra akan lebih mudah di proses pada proses berikutnya yaitu ekstraksi ciri. Ekstraksi ciri dilakukan agar memperoleh ciri-ciri khusus dari citra tersebut untuk selanjutnya akan digunakan dalam proses klasifikasi. Metode yang digunakan pada tahap ini adalah percobaan metode *Local Binary Pattern* (LBP). Proses berikutnya adalah proses klasifikasi dengan menggunakan metode *Support Vector Machine* (SVM) dan *K-Nearest Neighbor* (KNN). Proses ini merupakan lanjutan dari proses ekstraksi ciri dimana hasil dari ekstraksi ciri akan diolah menggunakan metode klasifikasi SVM dan KNN untuk mengidentifikasi kata atau kalimat dari aksara Bali tersebut.

3.2 Proses Pengenalan Aksara Bali

3.2.1 Akusisi Citra

Akusisi citra merupakan tahap awal dengan tujuan untuk mendapatkan citra sebagai data latih maupun data uji, yang nantinya citra tersebut akan diproses ke proses selanjutnya.

3.2.2 *Pre-processing*

Pre-processing merupakan proses yang dilakukan pada suatu citra sebelum dilakukan pemrosesan citra selanjutnya. Tujuan dari *pre-processing* adalah untuk meningkatkan kualitas dari citra masukan yang diperoleh. Adapun proses-proses di dalam *pre-processing* ini meliputi :

3.2.2.1 *Thresholding*

Proses *thresholding* digunakan untuk mengkonversi citra yang berwarna keabuan menjadi biner atau hitam putih. Nilai *threshold* (T) yang digunakan adalah 35% dari nilai keabuan suatu citra. Pemilihan nilai *threshold* ini ditentukan dari proses percobaan dengan nilai *threshold* 40% dan 30% dari nilai keabuan suatu citra, dimana pada percobaan dengan nilai *threshold* 0,4 *background* pada citra berubah warna menjadi warna putih, sedangkan pada percobaan dengan nilai *threshold* 0,30 piksel objek pada citra akan tersambung dengan piksel objek yang lain sehingga akan sulit untuk diklasifikasi. Formula yang digunakan dalam tahap *thresholding* dalam tugas akhir ini adalah

(4)

3.3.2.2 Erosi

Proses erosi diproses dengan matriks berukuran 9 x 9 yang bernilai 0 (hitam) dan 1 (putih) yang akan berjalan di sekitar citra untuk menghilangkan *noise* yang ada pada citra setelah melakukan proses *thresholding*. *Noise* yang dimaksud adalah adanya beberapa piksel yang berwarna putih pada citra tetapi bukan merupakan objek dari citra tersebut. Proses erosi menyebabkan piksel objek menjadi semakin tipis.

3.3.2.3 Dilasi

Proses dilasi diproses dengan matriks berukuran 9 x 9 yang bernilai 0 (hitam) dan 1 (putih) yang akan berjalan di sekitar citra untuk menebalkan piksel objek yang telah tipis karena proses erosi, sehingga menghasilkan output dengan objek piksel yang lebih fokus.

3.2.2.4 *Cropping*

Citra yang telah melalui proses dilasi, selanjutnya akan dilakukan proses *cropping*. Proses *cropping* merupakan proses yang digunakan untuk menghilangkan *background* sampai batas minimum dari objek, proses ini dilakukan agar citra hanya terdiri dari aksara Bali yang akan dikenali dan *background* citra juga berkurang sehingga tahap ekstraksi ciri menjadi lebih efektif.

3.2.2.5 *Resizing*

Proses *resizing* digunakan untuk pengubahan ukuran pada citra yang diperoleh dikarenakan citra hasil *cropping* memiliki ukuran yang bervariasi. Untuk keseragaman diperlukan ukuran data yang seragam dari data masukan dengan percobaan *resize* 400x600.

3.2.2.6 Segmentasi

Proses segmentasi dilakukan untuk mengetahui bagian – bagian pembentuk dari citra aksara Bali (dalam aplikasi ini adalah kata). Sehingga dapat membedakan antara kata yang satu dengan kata yang lainnya yang membentuk suatu kalimat dengan menggunakan aksara Bali.

3.2.3 Ekstraksi Ciri

Ekstraksi ciri adalah proses pengambilan ciri dari sebuah citra yang menggambarkan karakteristik dari suatu citra. Ciri yang didapatkan melalui proses ekstraksi ciri ini digunakan sebagai pembeda antara objek yang satu dengan objek yang lainnya. Proses dari metode LBP ini adalah melakukan *thresholding*, yaitu membandingkan piksel pusat dengan piksel tetangganya. Jika nilai piksel pusat lebih kecil atau sama dengan piksel tetangga maka akan diberikan nilai 1, sebaliknya jika piksel pusat lebih besar dari piksel tetangga maka akan diberikan nilai 0. Setelah melakukan proses *thresholding* maka dilakukan pembentukan pola biner sesuai urutan yang diperoleh dari rumus LBP, lalu diubah ke dalam bilangan desimal. Proses selanjutnya yaitu melakukan proses pembentukan histogram dari citra LBP tersebut, sehingga didapatkan vektor ciri dari histogram yang dibentuk.

3.2.4 Klasifikasi ciri

3.2.4.1 *Support Vector Machine*

Klasifikasi ini bertujuan untuk menentukan kelas yang tepat dari suatu citra uji berdasarkan ciri yang telah diekstraksi sebelumnya. Dengan menggunakan metode *one-againts-all* pada *multiclass* SVM, dibangun k(k-1)/2 buah model klasifikasi biner (k adalah jumlah kelas). Terdapat beberapa metode untuk melakukan pengujian setelah keseluruhan k(k-1)/2 model klasifikasi selesai dibangun. Salah satunya adalah metode *voting*. Setelah itu dilakukan proses dan perhitungan untuk menetukan *hyperplane* terbaik.

3.3.4.2 *K-Nearest Neighbor*

Pada Penelitian ini juga menggunakan metode klasifikasi KNN dengan jenis jarak *euclidean distance*. Pembangunan model K-NN terdiri dari ciri latih dan uji, ciri latih merupakan matriks berukuran MxN dimana M merupakan jumlah data dan N merupakan jumlah ciri masing-masing data. Untuk proses klasifikasi K-NN, model yang telah dibangun digunakan sebagai data acuan dari kelas sehingga ciri uji akan dihitung nilai jaraknya dengan persamaan jarak terhadap M data pada ciri latih. Analisis yang dilakukan pada K-NN adalah nilai *k* yang digunakan terhadap akurasi sistem. Nilai *k* yang digunakan saat pengujian adalah 1, 3, dan 5. Dipilihnya nilai *k* yang ganjil agar mengurangi kesalahan algoritma jika peluang kemiripannya sama.

3.3 Pengujian Sistem

Untuk pengujian sistem pengenalan aksara Bali ini digunakan parameter akurasi dan waktu komputasi. Akurasi adalah ukuran ketepatan sistem dalam mengenali masukan yang diberikan sehingga menghasilkan keluaran yang benar. Secara sistematis dapat dituliskan sebagai berikut:

$$\text{Akurasi} = \frac{\quad}{\quad} \qquad (5)$$

Waktu komputasi adalah waktu yang dibutuhkan sistem untuk melakukan suatu proses dalam satuan detik.

3.4 Hasil Pengujian Sistem dan Analisis

Hasil simulasi dapat dilihat pada gambar berikut :

Tabel 1 Akurasi pengaruh pola ketetangggan LBP dengan klasifikasi KNN

Pola Ketetanggan LBP	Jumlah Kata Terdeteksi Benar		
	k=1	k=3	k=5
[4,1]	32	30	29
[8,1]	32	33	32

Tabel 2 Akurasi pengaruh pola ketetangggan LBP dengan klasifikasi SVM

Pola Ketetanggan LBP	Jumlah Kata Terdeteksi Benar	
	Kernel= rbf	Kernel= poly
[4,1]	3	28
[8,1]	4	31

Tabel 3 Pencarian parameter ukuran *resize* yang paling optimal pada klasifikasi KNN

Ukuran *Resize*	Jumlah Kata Terdeteksi Benar		
	k=1	k=3	k=5
Fixed	36	37	37
Non Fixed	32	33	32

Tabel 4 Pencarian parameter ukuran *resize* yang paling optimal pada klasifikasi SVM

Ukuran *Resize*	Jumlah Kata Terdeteksi Benar	
	Kernel= rbf	Kernel= poly
Fixed	6	32
Non Fixed	4	31

Tabel 5 Pencarian pembagian citra yang paling optimal pada klasifikasi KNN **Tabel 6**

LBP	Jumlah Kata Terdeteksi Benar		
	k=1	k=3	k=5
Dibagi 4	46	47	45
Tidak Dibagi 4	36	37	37

Pencarian pembagian citra yang paling optimal pada klasifikasi SVM

LBP	Jumlah Kata Terdeteksi Benar	
	Kernel= rbf	Kernel= poly
Dibagi 4	4	39
Tidak Dibagi 4	6	32

Tabel 7 Waktu komputasi sistem

Pola Ketetanggaan LBP	Metode Klasifikasi	Waktu Komputasi Total Sistem	Waktu Komputasi Rata-Rata Sistem
[8,1]	KNN	208,82 detik	2,32 detik
	SVM	254,78 detik	4,24 detik
[4,1]	KNN	198,86 detik	2,20 detik
	SVM	249,08 detik	4,15 detik

4. Kesimpulan

Berdasarkan hasil pengujian simulasi dan implementasi yang dilakukan pada sistem pengenalan aksara Bali, menggunakan metode ekstraksi ciri LBP, serta metode klasifikasi SVM dan KNN memiliki kesimpulan sebagai berikut:

1. Sistem yang dirancang dengan metode ekstraksi ciri LBP dapat digunakan untuk mengenali aksara Bali dengan akurasi tertinggi adalah 74,6%, dengan menggunakan metode klasifikasi KNN.
2. Pada proses *resizing* dengan ukuran 400X600 piksel terbukti dapat menaikkan akurasi sistem.
3. Pola ketetanggan LBP [8,1] menghasilkan akurasi yang lebih baik daripada pola ketetanggan [4,1].
4. Proses pembagian citra terbukti dapat menghasilkan akurasi yang lebih baik dibandingkan dengan tidak melakukan proses pembagian citra.
5. Metode klasifikasi *K-Nearest Neighbor* menghasilkan akurasi yang lebih tinggi dibandingkan dengan metode klasifikasi *Support Vector Machine* . Terbukti pada saat nilai k = 3 dapat menghasilkan akurasi tertinggi. Hal tersebut dikarenakan dengan menggunakan metode SVM kurang cocok untuk klasifikasi banyak kelas dan memiliki ciri yang hampir sama.

DAFTAR PUSTAKA

[1] **"***Aksara Bali Digilas Modernisasi*." http://www.tobapos.com/2014/02/aksara-bali-digilas-modernisasi.html Tanggal Akses 13 Maret 2014

[2] Pietikäinen. M, Hadid, et al. 2010. *ComputerVision Using Local Binary Patterns*. London: Springer.

[3] Rachma, Annisa Dyah .2014. *Deteksi Kualitas Biji Kedelai Sebagai Bahan Baku Tempe Melalui Pengolahan Citra Digital dengan Ekstraksi Ciri LBP dan Metode Klasifikasi SVM*. Universitas Telkom. Bandung.

[4] Santosa, Budi. *Tutorial Support Vector Machine*.Institut Teknologi Sepuluh November

[5] "*Support Vector Machine (SVM)*." http://digilib.ittelkom.ac.id/index.php?view=article&catid=15%3Apemrosesan-sinyal&id=687%3Asvm&tmpl=component&print=1&page=&option=com_content&Itemid=14 diakses Tanggal Akses 15 Maret 2014

[6] S. Liao, M. W. K. Law, and A. C. S. Chung. 2009. *"Dominant local binary patterns for texture classification"* IEEE Trans. on Image Processing, vol. 18, no. 5, pp. 1107-1118.

[7] W. Hidayat. 2009. *Penerapan K-Nearest Neighbor Untuk Klasifikasi Gambar Landscape Berdasarkan Fitur Warna dan Tekstur*. Politeknik Telkom Bandung. Bandung

[8] Yuliaratih, Kadek Ayu Silvia. 2010. *Identifikasi Aksara Bali ke Huruf Latin dengan Menggunakan Klasifikasi Template Matching dan K-NN*. Institut Teknologi Telkom. Bandung.