Pengenalan Huruf Isyarat Tangan Menggunakan Ekstraksi Ciri *Local Binary Pattern*

M.Adhi Satria ^{#1}, Kurniawan Nur Ramadhani ^{#2}, Anditya Arifianto ^{#3}

[#] Laboratorium Multimedia, Fakultas Informatika, Universitas Telkom
Jl. Telekomunikasi No. 01, Terusan Buah Batu, Bandung 40257

¹ adhistr@student.telkomuniversity.ac.id

² kurniawannr@telkomuniversity.ac.id

³ anditya@telkomuniversity.ac.id

Abstract

In this research, we built a hand gesture recognition system using Local Binary Patterns (LBP) feature extraction method. The LBP method has been proved reliable in performing texture analysis, overcoming scaling and blurriness. For the classification algorithm, we used k-Nearest Neighbour (KNN) and Support Vector Machine (SVM) methods. The best LBP parameters are obtained for $R = 10$ and $P = 16$ using SVM with Gaussian kernel. The best performance in this research was obtained for F1-Score value of 99,84%.

Keywords: Sign Language, Local Binary Pattern, k-Nearest Neighbors, Support Vector Machine.

Abstrak

Pada penelitian ini dibangun sistem pengenalan huruf isyarat tangan menggunakan metode ekstraksi ciri Local Binary Patterns (LBP). Metode LBP memiliki kehandalan dalam melakukan analisis tekstur, mengatasi penskalaan dan citra yang kabur. Untuk algoritma klasifikasi, digunakan metode k-Nearest Neighbour (KNN) dan Support Vector Machine (SVM). Parameter LBP terbaik didapatkan untuk nilai $R=10$ dan $P=16$ menggunakan SVM dengan kernel Gaussian. Performansi terbaik dalam penelitian ini didapatkan untuk nilai F1-Score 99,84%.

Kata Kunci: Huruf Isyarat, Local Binary Patterns, k-Nearest Neighbors, Support Vector Machine.

I. PENDAHULUAN

Menurut hasil Survei Sosial Ekonomi Nasional (SUSENAS) pada tahun 2012 terdapat 6.008.661 orang penyandang disabilitas di Indonesia [4], dimana 472.855 orang merupakan tuna rungu wicara. Tuna rungu wicara dapat berkomunikasi menggunakan bahasa isyarat, akan tetapi masih banyak masyarakat yang belum bisa menggunakan bahasa isyarat. Oleh karena itu dibutuhkannya sistem pengenalan huruf isyarat.

Pada penelitian ini, dibangun sistem pengenalan huruf isyarat menggunakan metode ekstraksi ciri LBP, adapun metode LBP yang digunakan berupa LBP *non uniform pattern* dan LBP *uniform pattern*. Metode LBP dipilih dikarenakan andal dalam mengatasi *scaling* dan citra yang *blurred* [2]. Setelah dilakukan ekstraksi ciri, hasil ekstraksi ciri akan diklasifikasi menggunakan KNN dan SVM. KNN dipilih dikarenakan mudah diimplementasikan [3] dan SVM dipilih karena andal dalam mengklasifikasi data.

Penelitian mengenai pengenalan huruf isyarat telah banyak dilakukan dengan berbagai macam metode ekstraksi ciri dan klasifikasi. Di antara lainnya pada [13] dibangun sistem pengenalan huruf isyarat menggunakan *Histogram of Gradients* (HOG) dan *Principal Component Analysis* (PCA). Pada sistem yang telah dibangun, citra masukan akan disegmentasi menggunakan k-Means, kemudian hasil segmentasi akan diekstraksi ciri menggunakan metode HOG dan PCA, hasil performansi ekstraksi ciri HOG dan PCA akan

dibandingkan menggunakan metode klasifikasi KNN. Performansi ekstraksi ciri HOG mendapatkan akurasi 96%, lebih baik dibandingkan PCA dengan akurasi 88%. Lalu pada [5], telah dibuktikan bahwa metode *Gabor Filter* dan *Pyramidal Histograms of Oriented Gradients* (PHOG) juga dapat digunakan untuk pengenalan huruf isyarat dengan akurasi 98,75%. Selanjutnya dalam [6], dilakukan perbandingan antara metode *Local Binary Patterns* (LBP) dan *Linear Discriminant Analysis* (LDA), hasilnya LDA mendapatkan akurasi 92,417% yang mana kinerjanya lebih baik dibandingkan LBP dengan akurasi sebesar 87,13%. Kemudian pada [7], dilakukan reduksi dimensi data menggunakan PCA, lalu diklasifikasikan dengan metode *k-Nearest Neighbour* (KNN), kinerja sistemnya mendapatkan akurasi sebesar 97%. Selanjutnya pada [8], penelitian menggunakan metode *Hidden Markov Model* (HMM) digunakan untuk mengenali huruf isyarat secara *real-time*, akurasi yang didapat sebesar 96,5%.

II. DASAR TEORI

A. Local Binary Patterns

LBP merupakan metode yang sederhana dan handal untuk melakukan analisis tekstur [1]. Metode ini biasa digunakan sebagai *face descriptor*. LBP akan menentukan nilai parameter P dan R terlebih dahulu, dimana P ialah jumlah tetangga yang dibandingkan dan R merupakan radius jarak dengan tetangganya [9]. Metode ini akan membandingkan nilai pusat piksel dengan nilai piksel tetangga searah jarum jam untuk mendapatkan kode biner. Bila nilai piksel tetangga P lebih kecil dibandingkan dengan nilai pusat piksel, maka angka biner yang didapat ialah nol. Sebaliknya, bila nilai titik pusat piksel lebih atau sama dengan nilai tetangga, maka nilai biner yang didapat yaitu satu. Seperti pada persamaan 1.

$$b_j = \begin{cases} 1, & \text{jika } p(x_i, y_i) \geq p(x_0, y_0) \\ 0, & \text{selainnya} \end{cases} \quad (1)$$

Nilai $p(x,y)$ adalah nilai pixel pada posisi koordinat (x,y) pada citra. Contoh penerapan LBP $P=8$ dan $R=1$ dapat dilihat pada gambar 1. Pada gambar tersebut nilai titik pusat piksel ialah lima. Sehingga bila nilai piksel tetangga lebih kecil dari lima, maka nilai biner yang didapat ialah nol, sebaliknya nilai biner yang didapat yaitu satu.

Gambar 1 Contoh Penerapan LBP [9]

B. K-Nearest Neighbors (KNN)

KNN merupakan algoritma *supervised learning* yang sederhana dan biasa digunakan sebagai *classifier*. KNN mengklasifikasikan sebuah data uji berdasarkan data latih terdekat. Terdapat beberapa *distance matrix* yang dapat digunakan untuk menghitung kedekatan (*similarity*) antara data, diantaranya *euclidean distance*, *manhattan distance* dan *minkowski distance*. Pengukuran *similarity* yang sering digunakan pada KNN yaitu *euclidean distance*. Adapun persamaan *euclidean distance* dapat dilihat pada persamaan 2.

$$d_E(p, q) = \sqrt{\sum_{i=1}^N (p_i - q_i)^2} \quad (2)$$

Pada persamaan 2 nilai p_i merupakan nilai data uji, q_i adalah nilai data latih dan N adalah dimensi dari data. Setelah menghitung jarak antara data uji dan data latih, lalu dilakukan pemilihan kelas untuk data uji berdasarkan kedekatan data dan nilai K [3].

Gambar 2 Contoh Penerapan KNN

Gambar 2 ialah salah satu contoh pengklasifikasian data menggunakan metode KNN. Pada gambar 2 nilai $K=3$, sehingga kelas data uji akan ditentukan berdasarkan tiga kelas data latih terdekat. Pada penelitian ini KNN akan digunakan sebagai metode klasifikasi.

C. Support Vector Machine (SVM)

Support Vector Machine (SVM) merupakan salah satu *classifier supervised learning* yang biasa digunakan. Tujuan dari SVM yaitu menghasilkan model *classifier* berdasarkan data latih yang mana akan memprediksi data uji. SVM akan menentukan nilai *margin* yang maksimal untuk memisahkan tiap kelas data [10]. Untuk memisahkan kelas data akan digunakan *hyperplane*. Jarak antara *hyperplane* dengan data terdekat disebut dengan *margin* dan *hyperplane* dengan performansi terbaik yaitu *hyperplane* dengan nilai *margin* terbesar, hal ini disebut dengan *maximal margin hyperplane* [11]. Untuk menentukan *hyperplane* dapat menggunakan persamaan 3, dimana y_i merupakan kelas data, w bobot vektor dan b merupakan bias.

$$y_i(w * x_i + b) \geq 1, i = 1..N \quad (3)$$

Tidak hanya *maximal margin hyperplane*, namun pada SVM juga terdapat *soft margin classifier*, dimana SVM memiliki sebuah koefisien C yang berfungsi untuk memperbolehkan kesalahan *margin* [10]. *Soft margin classifier* digunakan agar *hyperplane* tidak terlalu kaku. Dengan menggunakan koefisien C maka kompleksitas model SVM yang akan dibangun meningkat. Adapun pengaruh koefisien C pada SVM.

1. Semakin kecil nilai koefisien C , maka semakin tinggi varians data, akan tetapi bias semakin kecil.
2. Semakin besar nilai koefisien C , maka semakin rendah varians data, dan bias semakin besar.

Sebelum menerapkan metode SVM sebaiknya data dilakukan normalisasi terlebih dahulu. Hal ini untuk menghindari perhitungan yang sulit dan menghindari atribut dengan rentang data lebih besar mendominasi atribut rentang data yang lebih kecil. Pada penelitian ini SVM digunakan sebagai *classifier*.

III. PERANCANGAN SISTEM

Pada penelitian ini dibangun sebuah sistem yang dapat mengenali huruf isyarat menggunakan ekstraksi ciri LBP. LBP yang digunakan berupa LBP *non uniform* dan LBP *uniform*. Sedangkan untuk proses klasifikasi menggunakan *classifier* SVM dan KNN. Agar sistem dapat mengenali huruf isyarat dengan tepat terdapat dua skema yang dilakukan. Skema pertama merupakan skema pembentukan model (*modelling*) sistem. Skema

modelling ini bertujuan untuk melakukan ekstraksi ciri pada data latih dan membangun *classifier* yang sesuai dengan data ekstraksi ciri. Pada skema *modelling* masukan berupa citra, dimana setiap citra mengalami proses *preprocessing* dan ekstraksi ciri. Setelah seluruh ekstraksi ciri citra didapatkan, maka dibangun model *classifier* yang tepat untuk mengklasifikasikan citra. Adapun skema *modelling* dapat dilihat pada gambar 3.

Gambar 3 Skema Modelling

Skema kedua yaitu skema pengujian, skema ini bertujuan untuk melakukan pengujian terhadap sistem yang telah dibangun pada skema *modelling*. Skema ini tidak jauh berbeda dengan skema *modelling*, tetapi pada skema ini *classifier* yang dibangun digunakan untuk mengklasifikasikan data uji yang ada. Skema pengujian dapat dilihat pada gambar 4.

Gambar 4 Skema Pengujian

IV. HASIL PENGUJIAN DAN ANALISIS

A. Skenario 1

Skenario ini bertujuan untuk menentukan nilai parameter P, R, dan *pattern* terbaik LBP dalam mengenali huruf isyarat. *Classifier* yang digunakan pada skenario ini yaitu KNN dengan nilai K=1 dan SVM dengan kernel

gaussian. Nilai P yang diujikan yaitu P=8 dan P=16, sedangkan nilai R yang diobservasi yaitu R=1, R=2, R=3, R=4, R=5, R=6, R=7, R=8, R=9, R=10. Ketika nilai P=16, observasi pada R=1 tidak bisa dilakukan karena nilai minimum R pada P=16 yaitu R=2. Berikut merupakan hasil pengujian yang dilakukan.

Gambar 5 Nilai F-1 Score untuk LBP Non Uniform P=8

Untuk observasi LBP *non uniform pattern* P=8 dapat dilihat pada gambar 5. Dari gambar dapat dilihat bahwa semakin besar nilai R maka performansi semakin meningkat, baik menggunakan metode KNN ataupun SVM. Performansi terbaik didapat ketika P=8 R=10 menggunakan *classifier* SVM dengan *F1-Score* 96,06%.

Gambar 6 Nilai F-1 Score untuk LBP Uniform P=8

Untuk pengujian LBP *uniform pattern* P=8 dapat dilihat pada gambar 6. Pada pengujian ini dapat dilihat bahwa R=10 juga memiliki performansi terbaik, seperti pada pengujian yang dilakukan sebelumnya. Adapun

classifier SVM lebih handal dibandingkan dengan *classifier* KNN. *F1-Score* performansi terbaik pada LBP *uniform pattern* P=8 yaitu 96,06%.

Gambar 7 Nilai *F1-Score* untuk LBP Non Uniform P=16

Untuk observasi LBP *non uniform pattern* P=16 dapat dilihat pada gambar 7. Berdasarkan gambar tersebut dapat dilihat bahwa semakin besar nilai R performansi semakin meningkat, sama seperti observasi yang dilakukan pada P=8. Akan tetapi pada observasi ini performansi KNN dapat mengungguli SVM, hal ini dikarenakan kernel *gaussian* tidak dapat mengatasi data yang memiliki dimensi tinggi [12], sedangkan LBP *non uniform pattern* P=16 memiliki 65.536 dimensi. Adapun *F1-Score* ketika P=16 dan R=10 menggunakan KNN yaitu 91,1%.

Gambar 8 Nilai *F1-Score* untuk LBP Uniform P=16

Untuk pengujian LBP *uniform pattern* P=16 dapat dilihat pada gambar 8. Performansi terbaik didapat ketika nilai R=10 dengan nilai *F1-Score* mencapai 96,53% menggunakan *classifier* SVM. Dari seluruh observasi yang dilakukan dengan berbagai nilai P dapat dilihat bahwa nilai R=10 memiliki performansi yang lebih baik

dibandingkan nilai R lainnya. Kemudian bila dibandingkan dengan seluruh kombinasi parameter LBP yang dilakukan, LBP *uniform pattern* dengan $P=16$ dan $R=10$ memiliki performansi terbaik. $R=10$ selalu memiliki performansi terbaik dengan setiap nilai P yang dicoba, hal ini dikarenakan semakin besar nilai R maka gambar hasil ekstraksi ciri semakin terlihat jelas, seperti pada gambar 9.

Gambar 9 Hasil Ekstraksi Ciri LBP $P=8$

B. Skenario 2

Skenario ini bertujuan untuk menentukan parameter KNN dan SVM terbaik dalam mengenali huruf isyarat. Pada KNN akan ditentukan nilai K yang memiliki performansi terbaik. Nilai K yang diuji pada metode KNN dimulai dari 1 hingga 9, dimana nilai K yang diuji hanya nilai yang berupa bilangan ganjil. Sedangkan observasi yang dilakukan pada metode SVM digunakan untuk mengetahui kernel SVM dengan performansi terbaik. Terdapat tiga kernel yang diuji yaitu, *polynomial*, *gaussian*, dan *linear*.

Gambar 10 Nilai F1-Score untuk beberapa nilai K pada algoritma KNN

Pada gambar 10 dapat dilihat perbandingan performansi KNN dengan nilai K yang berbeda, dapat dilihat bahwa semakin besar nilai K maka performansi akan semakin menurun, hal ini dikarenakan kemiripan data yang tinggi, sehingga semakin besar nilai K maka performansi *F1-Score* menurun.

Gambar 11 Nilai *F1-Score* untuk beberapa Kernel SVM

Pengujian kernel SVM dapat dilihat pada gambar 11, dimana kernel *polynomial* memiliki performansi terbaik dibandingkan dengan kernel lainnya dengan *F1-Score* 98,11%. Setelah mencoba melakukan observasi pada kernel dan nilai K. Kemudian dilakukan percobaan dengan menormalisasi data ekstraksi ciri. Normalisasi yang digunakan berupa normalisasi data dengan rentang nilai 0-1. Adapun hasilnya dapat dilihat pada gambar 12.

Gambar 12 Perbandingan Performansi Classifier Data Normalisasi dan Tanpa Normalisasi

Berdasarkan gambar 12, dapat dilihat bahwa normalisasi data tidak mengubah performansi *classifier* SVM. Sedangkan untuk KNN dengan melakukan normalisasi data dapat meningkatkan performansinya, *F1-Score* yang dicapai ketika data dinormalisasi mencapai 97,16% sedangkan bila data tidak dinormalisasi *F1-Score* hanya 87,71%.

C. Tahap Pengujian

Berdasarkan analisis hasil validasi, didapatkan kombinasi parameter LBP dengan performansi terbaik, yaitu LBP *uniform pattern* dengan nilai $P=16$ dan $R=10$ dan data ekstraksi ciri yang dinormalisasi. Selain itu juga diketahui bahwa kernel *polynomial* memiliki performansi terbaik dibandingkan dengan kernel SVM lainnya. Sedangkan nilai K untuk mendapatkan performansi terbaik yaitu $K=1$. Berikut merupakan hasil pengujian dengan menggunakan nilai parameter terbaik

Gambar 13 Hasil Pengujian Sistem

Berdasarkan gambar 13 dapat dilihat bahwa performansi KNN dan SVM memiliki performansi yang sama, dimana nilai *F1-Score* keduanya yaitu 99,84%, yang mana masing-masing *classifier* salah memprediksi satu gambar.

V. KESIMPULAN

Setelah melakukan pengujian dan analisis pada sistem pengenalan huruf isyarat menggunakan metode LBP dapat disimpulkan, bahwa:

1. LBP yang terbaik yaitu ketika parameter $P=16$ dan $R=10$, hal ini dikarenakan $P=16$ melakukan perbandingan titik pusat piksel ke tetangga lebih detail dibandingkan $P=8$. Sedangkan nilai $R=10$ dapat mencapai performansi lebih baik dibandingkan nilai R lainnya, karena semakin besar nilai R maka hasil ekstraksi ciri akan semakin terlihat jelas. Performansi *F1-Score* menggunakan LBP $P=16$ dan $R=10$ mencapai 87,71% menggunakan *classifier* KNN dan 96,53% menggunakan *classifier* SVM dengan kernel *gaussian*.
2. LBP *uniform* lebih baik dibandingkan dengan LBP *non uniform* ketika $P=16$ dan $R=10$ dengan selisih nilai *F1-Score* 13,39% ketika menggunakan *classifier* SVM dengan kernel *gaussian*, hal ini dikarenakan *classifier* SVM dengan kernel *gaussian* tidak handal dalam mengklasifikasi data yang memiliki dimensi tinggi, sedangkan LBP *non uniform* $P=16$ memiliki 65.536 dimensi.
3. Parameter K terbaik pada KNN yaitu ketika nilai $K=1$, hal ini dikarenakan kemiripan data yang tinggi, sehingga semakin besar nilai K maka performansi *F1-Score* semakin menurun. Performansi $K=1$ mendapatkan *F1-Score* 87,71%. Sedangkan pada SVM penggunaan kernel *polynomial* meningkatkan *F1-Score* dari 96,53% menjadi 98,19%.
4. KNN dengan data yang telah dinormalisasi dapat meningkatkan *F1-Score* sebesar 9,45% pada tahap validasi, dikarenakan ketika data dinormalisasi seluruh dimensi memiliki rentang data yang sama, sehingga kemiripan data lebih mudah untuk diukur. Adapun hasil akhir pada tahap pengujian dengan *F1-Score* 99,82% menggunakan KNN dan 99,82% menggunakan SVM.

DAFTAR PUSTAKA

- [1] Pietikainen, T. M. (2004). *TEXTURE ANALYSIS WITH LOCAL BINARY PATTERNS*. Oulu.
- [2] Youdong Ding, H. P. (2011). Recognition of Hand-Gestures Using Improved Local Binary Pattern. *Multimedia Technology (ICMT), 2011 International Conference on*, 3171-3174.
- [3] Bolandraftar, S. B. (2013). Application of K-Nearest Neighbor (KNN) Approach for Predicting Economic Events: Theoretical Background. *Journal of Engineering Research and Applications*, 605–610.
- [4] Jerry J. Tula, S. (2015, September 16). *Pelayanan Penyandang Disabilitas Dalam Menggunakan Berbagai Sarana Aksesibilitas*. Diambil kembali dari Direktorat Jenderal Rehabilitasi Sosial: <https://rehsos.kemsos.go.id/modules.php?name=News&file=article&sid=1890>

- [5] D.K. Vishwakarma, P. K. (2016). A Framework for Recognition of Hand Gesture in Static Postures. *International Conference on Computing, Communication and Automation (ICCCA2016)*.
- [6] Mahmood Jasim, M. H. (2015). Sign Language Interpretation using Linear Discriminant Analysis and Local Binary Patterns. *Informatics, Electronics & Vision (ICIEV)*, year={2015}.
- [7] Nitesh S. Soni, P. D. (2015). Online Hand Gesture Recognition & Classification for Deaf Dumb. *Inventive Computation Technologies (ICICT)*.
- [8] Xu-hui ZHANG, J.-j. W.-l. (2016). Improvement of Dynamic Hand Gesture Recognition Based on HMM Algorithm. *Information System and Artificial Intelligence (ISAI)*.
- [9] Timo Ojala, M. P. (2002). Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence* .
- [10] Olivier Chapelle, P. H. (1999). Support Vector Machines for Histogram-Based Image Classification. *IEEE Transactions on Neural Networks*.
- [11] Ratna Astuti Nugrahaeni, K. M. (2016). Comparative Analysis of Machine Learning KNN, SVM, and Random Forests Algorithm for Facial Expression Classification. *International Seminar on Application for Technology of Information and Communication*.
- [12] Chih-Wei Hsu, C.-C. C.-J. (2013). A Practical Guide to Support Vector Classification.
- [13] Idicula, D. M. (2014). Recognition of Hand Gestures of English Alphabets using HOG Method. *International Conference on Data Science & Engineering (ICDSE)*.