

PENGENALAN TANDA TANGAN MENGGUNAKAN *HISTOGRAM OF ORIENTED GRADIENTS* DAN *SMOOTH SUPPORT VECTOR MACHINE*

Ocky Marthatiyanda¹, Irfan Maliki, S.T, M.T²

^{1,2} Teknik Informatika-Universitas Komputer Indonesia

Jln. Dipatiukur No. 112-116, Lebakgede, Cobleng, Kota Bandung, Jawa Barat 40132

E-mail : ockymarthatiyanda@gmail.com¹, irfanmaliki007@gmail.com²

ABSTRAK

Tanda tangan merupakan ciri khas dari setiap orang sebagai bukti mengesahkan dokumen ataupun transaksi. Hal ini banyaknya pemalsuan tanda tangan, jika suatu transaksi bermasalah atau gagal karena adanya pada pemalsuan tanda tangan, tentu mengalami kerugian dimulai dari transaksi bermasalah ataupun gagal dalam mengesahkan dokumen-dokumen legal. Sehingga untuk mengatasi hal tersebut dapat dilakukan dengan cara mengidentifikasi tanda tangan dengan memanfaatkan teknik pengolahan citra. Untuk mengidentifikasi tanda tangan tersebut diperlukan beberapa metode di awal (*preprocessing*) seperti *cropping* dan *grayscale*. setelah proses *cropping* dilakukan untuk tahap selanjutnya dilakukan proses *grayscale* untuk inputan ekstraksi ciri dengan metode *Histogram Of Oriented Gradients* untuk mencari nilai fitur vektor dimana fitur vektor ini akan digunakan dalam mengklasifikasi beberapa label dari pemilik tanda tangan dengan metode *Smooth Support Vector Machine*. Pada penelitian ini digunakan data sebanyak 300 tanda tangan. Untuk pengujian menggunakan metode *k-fold cross validation* dimana *k* yang digunakan adalah *10-fold*. Berdasarkan hasil evaluasi pengujian pada pengenalan tanda tangan yang telah dilakukan dengan metode *k-fold cross validation* pada *10-fold* untuk semua skenario pengujian pada pemilik tanda tangan telah didapatkan nilai rata-rata akurasi sebesar 88%.

Kata kunci : *Signature Recognition, Image processing, handwriting, Histogram of Oriented Gradients, Smooth Support Vector Machine*

1. PENDAHULUAN

Tanda tangan merupakan tulisan tangan yang unik dan bersifat khusus sebagai substansi simbolik. Dalam tanda tangan biasa digunakan dalam kehidupan sehari-hari dan digunakan sebagai identifikasi seseorang yang fungsinya sebagai pembuktian dari pemilik tanda tangan, pada sebuah dokumen menyatakan bahwa pihak yang menandatangani, mengetahui dan menyetujui isi dari suatu dokumen. Pada penelitian sebelumnya tentang

offline signature recognition yang menggunakan ekstraksi cirinya *Grid Feature Segmentation Image* dan klasifikasinya menggunakan *Back Propagation* didapatkan akurasi sebesar 63%, karena hasil akurasi yang diperoleh memiliki nilai kecil karena *preprocessing* rotasi gambar tidak dilakukan, sehingga tanda tangan gambar miring dengan derajat tertentu akan menghasilkan output dengan nilai yang sangat berbeda dan menghasilkan kesalahan dalam hasil klasifikasi. [1]. Penelitian selanjutnya pernah dilakukan tentang *Offline Handwritten Signature Verification* menggunakan fitur *C-program* dan *Associative Memory Net (AMN)* dengan memberikan akurasi sebesar 92.3%, karena data yang digunakan adalah 12 tanda tangan [2]. Mengacu pada penelitian tentang *handwritten character* menggunakan fitur HOG dan klasifikasi menggunakan *Support Vector Machine (SVM)* bisa mendapatkan akurasi sebesar 96,56% [3]. Penelitian tentang Klasifikasi Pasien Diabetes Mellitus Menggunakan Metode *Smooth support vector machine* didapatkan akurasi sebesar 95,12% [4].

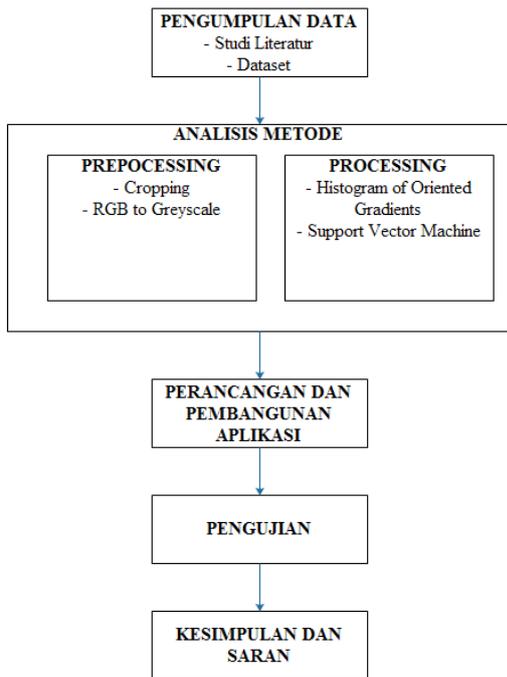
Berdasarkan penjelasan yang telah dipaparkan diatas, maka penulis tertarik untuk mengambil topik penelitian tugas akhir dengan judul “Pengenalan Tanda Tangan Menggunakan *Histogram of Oriented Gradients* dan *Smooth Support Vector Machine*”.

2. ISI PENELITIAN

Pada bagian ini membahas tentang metode penelitian, landasan teori, analisis sistem, dan hasil pengujian.

2.1 Metode Penelitian

Metode penelitian yang dilakukan menggunakan metode penelitian berdasarkan studi kasus. Studi kasus, atau penelitian kasus (*Case study*), adalah penelitian status subjek berdasarkan penelitian yang berkenaan dengan khas dari keseluruhan personalitas [5]. Adapun langkah – langkah penelitian seperti yang terlihat pada Gambar 1



Gambar 1 Langkah – langkah penelitian

2.2 Citra Digital

Secara umum, istilah pengolahan citra digital menyatakan “pemrosesan gambar berdimensi-dua melalui komputer digital. Foto adalah contoh gambar berdimensi dua yang dapat diolah dengan mudah. Setiap foto dalam bentuk citra digital (misalnya berasal dari kamera digital) dapat diolah melalui perangkat tertentu. Sebagai contoh, apabila hasil bidikan kamera terlihat agak gelap, citra dapat diolah menjadi lebih terang dimungkinkan pula untuk memisahkan foto orang dari latar belakangnya. Gambaran tersebut menunjukkan hal sederhana yang dapat dilakukan melalui pengolahan citra digital. Tentu saja banyak hal pelik lain yang dapat dilakukan melalui pengolahan citra digital [6].

2.3 Grayscale

Citra *grayscale* menangani gradasi warna hitam dan putih yang menghasilkan efek warna abu-abu. Gradasi warna pada setiap gambar dinyatakan dengan intensitas. Dalam hal ini, intensitas berkisar diantara 0 sampai dengan 255. Nilai 0 menyatakan hitam dan nilai 255 menyatakan putih. Pada grayscale nilai intensitas tersebut dapat diseragamkan dengan suatu fungsi. Berikut ini adalah rumus konversi citra berwarna (RGB) menjadi nilai intensitas *grayscale* [7].

$$G = wR.Red + wG.Green + wB.Blue \quad (1)$$

Keterangan :

$G(x, y)$ = Matriks *grayscale* koordinat piksel.

Red = Nilai merah (*Red*) dari suatu titik piksel

Green = Nilai hijau (*Green*) dari suatu titik piksel

Blue = Nilai Biru (*Blue*) dari suatu titik piksel

wR, wG, wB = Bobot setiap nilai elemen warna.

Persamaan 1 merupakan salah satu rumus yang digunakan untuk mengkonversi citra berwarna menjadi *grayscale*. Pada citra *grayscale*, setiap piksel adalah representasi derajat intensitas atau keabuan. Terdapat 256 jenis derajat keabuan pada citra *grayscale*. Mulai dari putih, kemudian semakin gelap sampai warna hitam. Oleh karena terdapat kemungkinan 256 derajat keabuan, maka setiap piksel pada *grayscale* disimpan 1 byte dalam memory (8 bits).

2.4 Histogram of oriented Gradients

Histogram of oriented Gradients merupakan cara umum untuk memperoleh deskriptor untuk deteksi objek tertentu. Sebagai contoh, untuk mendeteksi keberadaan manusia (*Human detection*) seperti penelitian yang pernah dilakukan oleh Dalal dan Trigs [8]. Proses dari algoritma ini dapat dijelaskan sebagai berikut :

1. *Preprocessing*, melakukan intensitas normalisasi atau dengan kata lain mengubah citra gambar menjadi *greyscale* (Citra berskala keabuan).
2. Menghitung *edge map*. Mengestimasi arah x dan y pada gambar lalu menghitung gradien *magnitudes* dan gradien *angle* untuk setiap pixel gambar. Berikut ini merupakan rumus perhitungan untuk gradien *magnitudes* dan gradien *angle* :

$$|G| = \sqrt{I_x^2 + I_y^2} \quad (2)$$

Dimana $|G|$ adalah bilangan absolut dari gradien *magnitudes* dan I adalah citra gambar yang sudah *greyscale* dari hasil tahapan sebelumnya. I_x merupakan matrik terhadap sumbu-x dan I_y merupakan matrik terhadap sumbu-y. I_x dan I_y dapat dihitung dengan rumus perhitungan sebagai berikut :

$$I_x = I * D_x \text{ dan } I_y = I * D_y \quad (3)$$

D_x adalah *mask* $[-1 \ 0 \ 1]$, sedangkan D_y adalah *mask* $\begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$ masing – masing dihitung dengan cara

konvolusi (tanda *). Kemudian gradien *angle* ke dalam koordinat sumbu dengan sudut diantara 0 sampai 180 dapat dihitung dengan rumus perhitungan sebagai berikut :

$$\theta = \arctan\left(\frac{I_x}{I_y}\right) \quad (4)$$

3. *Spatial binning*. Tahapan selanjutnya adalah melakukan perhitungan *histogram* dari gradien *angle* ke tiap-tiap *cell*. Setiap pixel dalam sebuah *cell* mempunyai nilai *histogram* nya sendiri – sendiri berdasarkan nilai yang dihasilkan dalam perhitungan gradien yang kemudian dilakukan normalisasi pada setiap blok. *Cell* memiliki ukuran 8x8 pixel pada sebuah citra. Sedangkan blok memiliki ukuran 2x2 *cell*.
4. *Normalize voting values for generating a descriptor*. Nilai normalisasi fitur blok

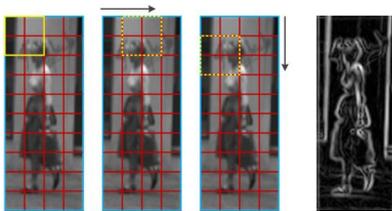
selanjutnya didapat dengan rumus perhitungan sebagai berikut [8] :

$$norm = \frac{v(n)}{\sqrt{(\sum_{k=1}^{block \cdot l} v(k)^2)+1}} \quad (5)$$

Nilai v merupakan nilai gradien *magnitudes* sedangkan n adalah jumlah *bins* dan *block* (2×2 cell) merupakan jumlah *cell* sedangkan l merupakan jumlah blok yang tidak *overlap*. Fitur blok dinormalisasi untuk mengurangi efek perubahan kecerahan obyek pada satu blok.

5. *Augment all block vectors consecutively.* Setelah fitur blok dinormalisasi, nilai normalisasi setiap blok akan digabungkan menjadi satu vektor (vektor 1 dimensi) hasil satu vektor inilah bisa disebut sebagai fitur vektor *Histogram of oriented Gradients*.

Berikut ini merupakan ilustrasi dari cara kerja *Histogram of oriented Gradients* terdapat pada Gambar 2



Gambar 2 *Histogram of oriented Gradients*

2.5 Support Vector Machine

Ada dua cara untuk mengimplementasikan *multiclass* SVM yaitu dengan melakukan penggabungan beberapa SVM biner dan penggabungan semua data yang terdiri dari beberapa kelas ke dalam bentuk permasalahan optimasi. konsep SVM dapat dijelaskan secara sederhana sebagai usaha mencari *hyperplane* terbaik yang berfungsi sebagai pemisah dua buah class pada *input space*. Berikut metode yang umum digunakan untuk mengimplementasikan SVM dengan pendekatan sebagai berikut [9]:

1. Metode *one-against-all*

Dengan menggunakan metode ini, akan dibangun k buah model SVM biner (k adalah jumlah kelas). Contohnya, terdapat permasalahan klasifikasi dengan 4 buah kelas. Untuk pelatihan digunakan 4 buah SVM biner seperti pada Tabel 1 dan penggunaannya dalam mengklasifikasi kelas pada data baru dapat dilihat pada persamaan sebagai berikut [10]:

$$Kelas\ x = \arg \max_{i=1..k} ((w^{(i)})^T \cdot \phi(x) + b^{(i)}) \quad (6)$$

Dengan menentukan *hyperplane* terbesar pada nilai x maka akan mengklasifikasikan kelas tersebut.

Tabel 1 4 SVM biner metode *One-against-all*

$y_i = 1$	$y_i = -1$	Hipotesis
Kelas 1	Bukan kelas 1	$f^1(x) = (w^1)x + b^1$
Kelas 2	Bukan kelas 2	$f^2(x) = (w^2)x + b^2$

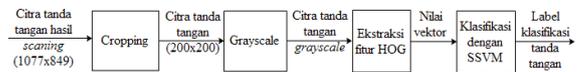
Kelas 3	Bukan kelas 3	$f^3(x) = (w^3)x + b^3$
Kelas 4	Bukan kelas 4	$f^4(x) = (w^4)x + b^4$

2.6 Smooth Support Vector Machine

SSVM adalah pengembangan SVM dengan menggunakan teknik *smoothing*. Metode ini pertama kali diperkenalkan oleh Lee pada tahun 2001. SVM memanfaatkan optimasi dengan *quadratic programming*, sehingga untuk data berdimensi tinggi dan data jumlah besar SVM menjadi kurang efisien. Oleh karena itu dikembangkan *smoothing technique* yang menggantikan *plus function* SVM dengan integral dari fungsi sigmoid neural network yang selanjutnya dikenal dengan *Smooth Support Vector Machine* (SSVM) [11].

2.7 Analisis Sistem

Pada bagian ini untuk menjelaskan proses alur penelitian agar lebih mudah dipahami. Hal ini dapat dilihat pada gambaran alur sistem pada Gambar 3



Gambar 3 Gambaran Alur Sistem

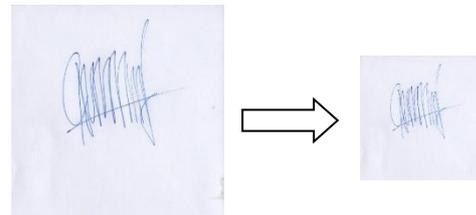
Penjelasan dari tahapan – tahapan diatas adalah sebagai berikut :

1. Data Citra tanda tangan

Merupakan tahapan awal citra yang berasal dari hasil *scanning* tanda tangan berupa file .jpg.

2. *Cropping*

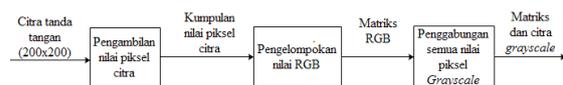
Merupakan proses memotong citra tanda tangan dengan size 200x200 piksel dan *cropping* digunakan adalah teknik *resize*. perubahan sebelum di *cropping* dan setelah di *cropping* seperti pada Gambar 4



Gambar 4 Contoh citra telah di *Cropping*

3. *Grayscale*

Grayscale digunakan untuk mempermudah proses lanjutan dengan menyederhanakan nilai pada matriks ke nilai warna putih dan abu-abu untuk ke proses ekstraksi ciri. Berikut proses perubahan citra ke *grayscale* seperti pada Gambar 5



Gambar 5 Proses *Grayscale*

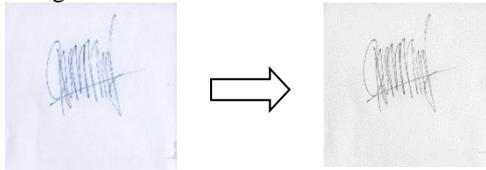
Contohnya citra pada piksel 80,120 memiliki nilai warna $Red=83$, $Green=73$ dan nilai $Blue=73$ maka berdasarkan persamaan 1 adalah sebagai berikut :

$$G(x, y) = wR.Red + wG.Green + wB.Blue$$

$$\begin{aligned}
G(0,0) &= (0,299 \text{ Red}) + (0,587 \times \text{Green}) \\
&\quad + (0,114 \times \text{Blue}) \\
&= (0,299 \times 84) + (0,587 \times 73) \\
&\quad + (0,114 \times 75) \\
&= 25,12 + 42,85 + 8,55 \\
&= 76,52
\end{aligned}$$

Dari perhitungan di atas, nilai piksel 76,52 dibulatkan ke atas menjadi 77.

Contoh hasil dari gambar sebelum di *grayscale* dan sesudah di *grayscale* bisa dilihat pada Gambar 6 sebagai berikut :



Gambar 6 Contoh hasil *grayscale*

4. Ekstraksi fitur HOG

Merupakan proses untuk mendapatkan ekstraksi ciri tiap gambar yang akan dijadikan data latih. Sebagai contoh untuk tahapan ini akan diambil sebuah sampel data latih citra RGB matriks berukuran 200 x 200 yang dapat dilihat pada Gambar 7 sebagai berikut :



Gambar 7 Sampel data latih ukuran 200 x 200

Berikut ini merupakan proses metode HOG :

a. Menghitung *edge map*

Pada proses ini akan dilakukan perhitungan untuk mencari nilai *gradien magnitudes* dan *angle* untuk setiap pixel gambar yang sudah berskala keabuan (*Grayscale*) sebelum itu akan dilakukan konvolusi

dengan *mask* D_x yaitu $[-1 \ 0 \ 1]$ dan D_y yaitu $\begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$

untuk mencari nilai I_x dan I_y dengan matriks *Grayscale* berukuran 3 x 3 sebagai sampel untuk dicari nilai *gradien magnitudes* dan *angle* seperti persamaan 3 sebagai berikut :

$$\begin{aligned}
I_x &= I * D_x \text{ dan } I_y = I * D_y \\
I_x &= \begin{bmatrix} 77 & 84 & 74 \\ 131 & 121 & 88 \\ 239 & 98 & 176 \end{bmatrix} * [-1 \ 0 \ 1] \\
I_y &= \begin{bmatrix} 77 & 84 & 74 \\ 131 & 121 & 88 \\ 239 & 98 & 176 \end{bmatrix} * \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}
\end{aligned}$$

Sebagai contoh perhitungan konvolusi hanya pada koordinat (1,2) untuk I_x dan I_y maka hasilnya adalah :

$$\begin{aligned}
I_x &= |(77 * -1) + (84 * 0) + (74 * 1)| \\
I_x &= 3 \\
I_y &= |(84 * -1) + (121 * 0) + (98 * 1)| \\
I_y &= 14
\end{aligned}$$

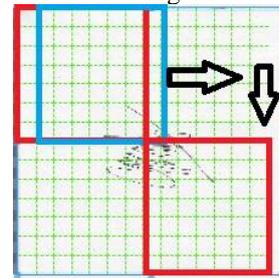
Selanjutnya akan dilakukan perhitungan *gradien magnitudes* dan *angle* seperti persamaan 2 dan 4 sebagai berikut :

$$\begin{aligned}
|G| &= \sqrt{I_x^2 + I_y^2} \\
\theta &= \arctan\left(\frac{I_x}{I_y}\right) \\
|G| &= \sqrt{3^2 + 14^2} \\
|G| &= \sqrt{9 + 196} \\
|G| &= 14 \text{ (Dibulatkan)} \\
\theta &= \arctan\frac{9}{196} \\
\theta &= \arctan(0,04) \sim 2,3^\circ
\end{aligned}$$

Maka didapatkan nilai *gradien magnitudes* dan *angle* yaitu 14 dan $2,3^\circ$ untuk koordinat (1,2).

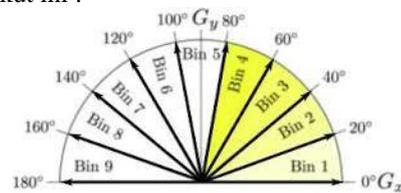
b. Melakukan *Spatial binning*

Pada proses ini akan dilakukan perhitungan *histogram* dari *gradien angle* yang sudah didapatkan diproses sebelumnya ke tiap – tiap *cell*. Setiap pixel dalam sebuah *cell* akan mempunyai nilai *histogram* nya sendiri – sendiri berdasarkan nilai yang dihasilkan dalam perhitungan *gradien* yang kemudian dilakukan normalisasi pada setiap blok. *Cell* untuk penelitian ini memiliki ukuran 7 x 7 yang akan terus bergerak ke arah kanan dan kebawah sampai *cell* berada pada ujung gambar seperti yang terlihat pada Gambar 8 sebagai berikut :



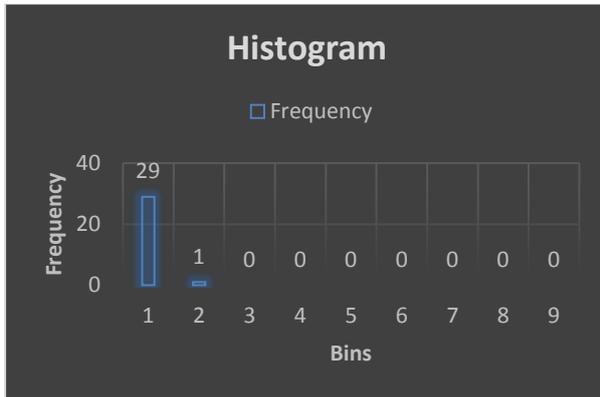
Gambar 8 *Cell* bergerak ke ujung gambar

Berikut ini merupakan contoh 9 *bins* dengan orientasi 0 - 180° seperti yang terlihat pada Gambar 17 berikut ini :



Gambar 9 Contoh 9 *bins* orientasi 0- 180°

Hasil dari pemberian semua *voting* dalam bentuk sebuah *histogram* dari citra data latih terdapat pada Gambar 10 sebagai berikut :



Gambar 10 Histogram citra data latih

c. *Normalize voting values for generating a descriptor*

Langkah selanjutnya akan dilakukan perhitungan normalisasi nilai fitur blok pada setiap nilai *gradient magnitude* pada proses sebelumnya di *spatial binning* maka rumus perhitungannya seperti persamaan 11 sebagai berikut:

$$\begin{aligned}
 norm &= \frac{v(n)}{\sqrt{(\sum_{k=1}^{block \times l} v(k)^2) + 1}} \\
 &= \frac{88(9)}{\sqrt{(\sum_{k=1}^{7 \times 7 \times 16} 88(k)^2) + 1}} \\
 &= \frac{792}{\sqrt{(\sum_{k=1}^{784} 88(k)^2) + 1}} \\
 &= \frac{792}{\sqrt{14162505281}} \\
 &= 0,006655
 \end{aligned}$$

Normalisasi nilai fitur blok dilakukan bertujuan untuk mengurangi efek perubahan kecerahan obyek pada setiap blok.

5. Klasifikasi dengan SSVM

Proses klasifikasi adalah untuk memilih keputusan yang akan diambil terkait dengan pemilik tanda tangan yang telah di *training* [12].

Berikut ini merupakan hasil fitur vektor HOG dari citra data latih dapat dilihat pada Tabel 2 sebagai berikut :

Tabel 2 Hasil fitur vektor HOG citra data latih

0: 0.09737 1	1: 0.03230 8	2: 0.10675 2	3: 0.02060 4	4: 0.29416 8
5: 0.00413 4	6: 0.03642 7	7: 0.07081 4	8: 0.10966 2	9: 0.04396 3
10: 0.00568 6	11: 0.01713 1	12: 0.02017 1	...	7055: 0.02394 3

a. Data latih SSVM

Merupakan proses untuk menyimpan hasil data latih SSVM yang nantinya akan dimuat ketika tahap

testing dimulai dan disimpan data *training* dalam bentuk XML.

b. Proses *testing*

Merupakan proses menguji citra tanda tangan dan akan dilakukan pencocokan dengan data latih SSVM yang tersimpan di XML.

c. Melakukan prediksi label

Merupakan proses prediksi SSVM untuk memprediksi angka yang keluar dari masukkan citra tanda tangan saat *testing*. sebagai contoh data uji yang sudah menemukan nilai *support vectornya* yaitu (0,10067,4.5) akan dimasukkan ke persamaan 6 sebagai berikut :

Kelas x

$$\begin{aligned}
 &= \arg \max_{k=1..3} \left(\begin{bmatrix} 31,33318391 \\ -61,05419883 \end{bmatrix} \right)^T \cdot \begin{bmatrix} 0,10067 \\ 4,5 \end{bmatrix} \\
 &+ (-7,43025373), \left(\begin{bmatrix} 17,36439644 \\ -27,50146985 \end{bmatrix} \right)^T \cdot \begin{bmatrix} 0,10067 \\ 4,5 \end{bmatrix} \\
 &+ (-2,534268352), \left(\begin{bmatrix} 20,09117293 \\ -31,78081239 \end{bmatrix} \right)^T \cdot \begin{bmatrix} 0,10067 \\ 4,5 \end{bmatrix} \\
 &+ (-2,922409864)
 \end{aligned}$$

Kelas x

$$= \arg \max_{k=1..3} (-278.8184968, -124.5428089, -143.9134872)$$

$$\text{Kelas } x = -124.5428089$$

Seperti yang bisa dilihat pada persamaan diatas nilai *hyperplane* yang paling besar adalah -124.5428089 yang dimana indeks nilai *hyperplane* tersebut adalah milik kelas/label 2.

2.8 Pengujian Akurasi

Pengujian akurasi pada penelitian ini dilakukan untuk mengetahui nilai akurasi, dalam implementasinya pengujian performansi menggunakan metode *k-fold cross validation*. *Cross validation* merupakan metode statistika untuk mengevaluasi dan membedakan data dengan membagi itu menjadi dua segmen, segmen pertama data *training* dan segmen kedua data *testing*. Biasanya *cross validation* membutuhkan data *training* dan *testing* yang harus *cross-over* pada setiap iterasinya. Pada dasarnya *cross validation* menggunakan metode *k-fold*. Pada *k-fold validation* data dibagi menjadi berapa banyak k , yang dimana k merupakan iterasi atau perulangan. [13].

Pengujian ini bertujuan untun menguji stabilitas akurasi jika diuji dengan data latih dan data uji yang berbeda. Pengujian ini data yang digunakan sebanyak 300 data yang dibagi menjadi 10 *subset* yang masing-masing berjumlah 30 data. Skenario uji performansi akurasi dengan metode *k-fold cross validation* dimana k yang digunakan adalah 10-*fold*.

Berdasarkan hasil pengujian keakuratan dari skenario pengujian yang telah dilakukan hasilnya dapat dilihat pada Tabel 3 sebagai berikut :

Tabel 3 Hasil akurasi skenario pengujian

Fold	Data Training	Data Testing	Akurasi
1	270	30	93.33 %
2	270	30	90.00 %
3	270	30	86.67 %
4	270	30	83.33 %
5	270	30	80.00 %
6	270	30	93.33 %
7	270	30	83.33 %
8	270	30	86.67 %
9	270	30	93.33 %
10	270	30	90.00 %
Rata-rata akurasi			88.00 %

Dari data tersebut dapat diambil kesimpulan bahwa hasil pengujian menggunakan *k-fold cross validation* dengan rata-rata akurasi sebesar 88%. Karena pada tahapan *preprocessing* tidak dilakukan proses segmentasi pada tanda tangan sehingga ada penurunan akurasi pada pengenalan tanda tangan.

3. PENUTUP

3.1 Kesimpulan

Kesimpulan dari penelitian ini yang berjudul: "Pengenalan tanda tangan menggunakan *Histogram of Oriented Gradients* dan *Smooth Support Vector Machine*" adalah sebagai berikut :

1. Hasil dari akurasi terhadap fitur *Histogram of Oriented Gradients* dan klasifikasi menggunakan *Smooth Support Vector Machine* pada pengenalan tanda tangan dengan metode pengujian *k-fold cross validation* diperoleh data akurasi sebesar 88%. Karena pada tahapan *preprocessing* tidak dilakukan proses segmentasi pada tanda tangan sehingga ada penurunan akurasi pada pengenalan tanda tangan.
2. Semakin bervariasi data tanda tangan yang digunakan dalam pelatihan maka lebih besar kemungkinan bisa mengenali secara tepat pada saat proses pengujian

3.2 Saran

Pada penelitian yang dikerjakan masih memiliki kekurangan, untuk itu dapat dilakukan pengembangan yaitu :

1. Perlunya dilakukan segmentasi pada bagian citra tanda tangan agar hasil lebih bagus pada pengenalan tanda tangan
2. Menggunakan metode RCNN untuk proses klasifikasi pada pengenalan tanda tangan untuk bisa mencapai akurasi yang tinggi

DAFTAR PUSTAKA

- [1] A. Rahmi, V. N. Wijyaningrum, W. F. Mahmudy, dan A. M. A. K. Parewe, "Offline signature recognition using back propagation neural network," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 4, no. 3, hal. 678–683, 2016.
- [2] T. Dash, T. Nayak, dan S. Chattopadhyay, "Offline Handwritten Signature Verification using Associative Memory Net," *Comput. Eng.*, vol. 1, no. 4, hal. 370–374, 2012.
- [3] R. L. Kulkarni, "International Journal of Advance Research in Handwritten Character Recognition Using HOG , COM by OpenCV & Python," *Int. J. Adv. Res. Comput. Sci. Manag. Stud. Res.*, vol. 5, no. 4, hal. 36–40, 2017.
- [4] R. A. Nugroho, Tarno, dan A. Prahutama, "Klasifikasi Pasien Diabetes Mellitus Menggunakan Metode Smooth Support Vector Machine," *Gaussian*, vol. 6, hal. 439–448, 2017.
- [5] M. Nazir, *Metode Penelitian*, 2009.
- [6] A. S. Abdul Kadir, *Teori dan Aplikasi Pengolahan Citra*, 2013.
- [7] S. R. Sternberg, *Grayscale Morphology, Computer vision, graphics, dan image processing*, 1986.
- [8] B. T. Navneet Dalal, "Histograms of Oriented Gradients for Human Detection," 2005. [Online]. Available: <https://lear.inrialpes.fr/people/triggs/pubs/Dala-l-cvpr05.pdf>. [Diakses 16 April 2018].
- [9] A. B. W. D. H. Anto Satriyo Nugroho, "Support Vector Machine Teori dan Aplikasinya dalam Bioinformatika," *IlmuKomputer.Com*, 2003.
- [10] P. A. Octaviani, "Penerapan Metode Klasifikasi Support Vector Machine (SVM) pada Data Akreditasi Sekolah Dasar (SD) di Kabupaten Magelang," Universitas Diponegoro, Semarang, 2014..
- [11] G. Indrawan, I. K. P. Sudiarsa, dan K. Agustini, "Smooth Support Vector Machine for Suicide-Related Behaviours Prediction," vol. 8, no. 5, hal. 3399–3406, 2018.
- [12] Widiastuti, Nelly Indiani. "Model Perilaku Berjalan Agen-Agen Menggunakan Fuzzy Logic." *Jurnal Komputer dan Informatika (KOMPUTA)* 1.1 (2012).
- [13] R. C. Sharma, K. Hara, dan H. Hirayama, "A Machine Learning and Cross-Validation Approach for the Discrimination of Vegetation Physiognomic Types Using Satellite Based Multispectral and Multitemporal Data," *Scientifica (Cairo)*, vol. 2017, hal. 1–8, 2017.

- [14] M. I. T. Taichiro Tokumori, "Histograms of Oriented Gradients(HOG)," Universitas Ryukyus, Jepang.