Jurnal Ilmiah Komputer dan Informatika (KOMPUTA)
Edisi. .. Volume. ... Bulan 20.. ISSN : 2089-9033

SUPPORT VECTOR MACHINE DALAM SISTEM PENDETEKSI KEPRIBADIAN BERDASARKAN POLA TANDA TANGAN

Bhakti Prasetya Utama[1], Ken Kinanti Purnamasari[2]

[1,2] Teknik Informatika – Universitas Komputer Indonesia
Jl. Dipatiukur 112-114 Bandung
Email : bhaktiprasetya95@gmail.com[1], ken.kinanti@email.unikom.ac.id[2]

ABSTRAK

Tanda tangan mempunyai pola tertentu berdasarkan fitur yang ditinjau dari ilmu grafologi untuk dapat mengungkapkan kepribadian penulisnya, namun saat ini sistem yang dibangun untuk mendeteksi kepribadian berdasarkan pola tanda tangan secara otomatis belum begitu baik untuk digunakan. Dalam penelitian ini akan mencoba menguji sistem dengan metode yang berbeda yaitu *support vector machine* dari beberapa penelitian sebelumnya untuk mengidentifikasi pola tanda tangan. Pengujian fitur berdasarkan kesembilan fitur dengan lima fitur yaitu awal kurva, coretan akhir, cangkang, coretan tengah dan garis bawah menggunakan *support vector machine* (SVM). Dan empat fitur lainnya yaitu margin ekstrim, struktur titik, tanda tangan terpisah, dan coretan garis terputus menggunakan metode *rulebased*. Pelatihan dengan metode SVM menggunakan 50 data latih untuk tiap fitur-fiturnya. Hasil identifikasi SVM terhadap 50 data uji tanda tangan pada tiap fiturnya memperoleh akurasi paling tinggi sebesar 89% dan seluruh rata-rata akurasi sebesar 60,7% akan tetapi hasil akurasi paling rendah sebesar 41%. Pada Metode *rulebased* diperoleh akurasi klasifikasi fitur tanda tangan paling tinggi sebesar 70% dan paling rendah sebesar 34% dengan seluruh rata-rata akurasi sebesar 52,8%.

Kata kunci : *Support Vector Machine*, SVM, pola tanda tangan, grafologi, *rulebased*.

1. PENDAHULUAN

Kepribadian seseorang dapat dideteksi melalui tulisan tangan. Deteksi kepribadian seseorang dengan tulisan tangan adalah seni atau ilmu yang disebut dengan *grafologi*[1]. Dalam *grafologi*, tulisan tangan dianalisis dengan elemen grafis struktural dalam rangka untuk memperoleh informasi tentang kepribadian penulis. Analisis ini didasari oleh 7 (tujuh) kategori yang dikemukakan oleh para cendekiawan asal eropa diabad ke-19. Hal tersebut merupakan dimensi, bentuk, tekanan, kecepatan, arah, tata letak dan kontinuitas[1]. Terbentuknya kategori ini menjadikan teori dan pedoman dalam menganalisis pola tulisan tangan. Demikian pula dengan tanda tangan yang merupakan bagian dari tulisan tangan. Tanda tangan juga dapat mencerminkan kepribadian seseorang berdasarkan pola tertentu.

Saat ini ilmu *grafologi* telah menarik banyak minat peneliti dalam melakukan pembangunan sistem berbasis komputer. Pada perkembangannya deteksi kepribadian dapat dikenali dengan memanfaatkan metode *machine learning*.

2. ISI PENELITIAN

2.1 Penelitian Terkait

Penelitian yang serupa sebelumnya pernah dilakukan oleh Esmeralda, Risna dan Nur Ramdlan untuk menganalisis pola tulisan tangan dalam mendeteksi kepribadian menggunakan metode *Multi Layer Perceptron* (MLP). Hasil dari penelitian tersebut diperoleh nilai akurasi sebesar 80%[2]. Penelitan serupa juga dilakukan oleh K.Nithya, A.Keerthana dan R.Lakshmi dengan menggunakan metode *fuzzy*. Dari penelitian ini didapatkan akurasi sebesar 86% [3]. Akan tetapi, kedua penelitian tersebut masih menghasilkan akurasi klasifikasi yang lebih kecil daripada penelitian yang dilakukan oleh Shitala, Vivek dan Akshay yang menggunakan metode *Support Vector Machine* (SVM). Dari penelitian mereka dihasilkan akurasi sebesar 90%[4].

Fokus utama penelitian ini yaitu dalam kasus analisis pola tanda tangan untuk mengetahui kepribadian penulisnya. Beberapa penelitian sebelumnya tentang kasus pola tanda tangan, yaitu penelitian yang dilakukan oleh Djamal dan Sheldy dengan menggunakan metode MLP serta penelitian oleh Thoyib yang menggunakan metode *Learning Vektor Quantization*(LVQ). Kedua penelitian tersebut mendapatkan akurasi sebesar 56-78% [8] dan 53%[9]. Dari penelitian-penelitian tersebut dapat dilihat bahwa hasil klasifikasi untuk analisis pola tanda tangan masih belum begitu baik. Oleh karena itu, akan dilakukan pengujian analisis pola tanda tangan yang diharapkan dapat meningkatkan kualitas akurasi klasifikasi.

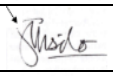
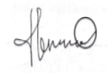
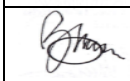
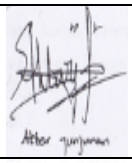
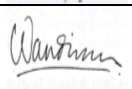
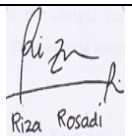
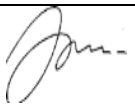
Berdasarkan penjelasan di atas, penelitian ini akan melakukan pengujian sistem pendeteksi

kepribadian berdasarkan pola tanda tangan dengan menggunakan metode *Support vector Machine* sebagai klasifikasinya.

2.2 Fitur Tanda Tangan Dalam Grafologi

Analisis tanda tangan berdasarkan sembilan fitur untuk menggambarkan apakah terdapat kemunculan kepribadian tertentu dari 15 tipe yang ditinjau. Untuk lebih jelas, diperlihatkan pada Tabel 1, yaitu pola tanda tangan beserta kepribadian yang menyertai.

Tabel 1 Fitur-Fitur Tanda Tangan[2]

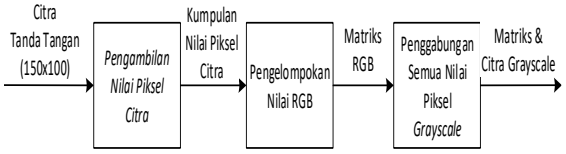
Fitur	Gambar	Ciri	Kepribadian
Cangkang		Lengkung Tertutup	Ketakutan berlebihan,introvert,tidak memperdulikan sekitar,tidak suka bergaul dan bekerja sama.
Awal kurva		Lengkung mundur	Nyaman akan masa lalu.
		Lengkung tajam	Mampu memformulasi pikiran secara tajam.
		Lengkung lembut	Hati-hati,ramah, diplomatis.
Coretan Akhir		Menaik	Terbuka,pandangan ke depan,keinginan maju, percaya diri.
		Menurun	Kurang semangat,berfikir realistis,kurang percaya diri,mudah putus asa.
Coretan Ditengah		Adanya coretan	Posesif
Garis Bawah		Adanya Garis bawah	Memiliki ide dan cara berpikir unik, membutuhkan dukungan membuat keputusan,serta memiliki keandalan dalam memimpin.
Margin ekstrim		Cenderung ke kanan	Ceroboh,kurang perhatian
		Cenderung ke kiri	Takut gagal,takut pada orang lain,kurang percaya diri,pesimis
		Cenderung di atas	Respek pada diri sendiri,mencerminkan pribadi bahagia.
		Cenderung ke bawah	Depresi,pemalu,merasa asing
Struktur titik		Ada titik	Pendirian stabil, memiliki rasa curiga,selalu menjaga jarak tidak mudah percaya.
Tanda Tangan Terpisah		Ada tanda tangan terpisah	Memiliki pengalaman kurang menyenangkan di masa lalu.
Garis terpisah		Ada garis terpisah	Membatasi keinginannya,tidak berani mengambil resiko,sering patah semangat dan ragu mengambil keputusan.

2.3 Citra Digital

Citra digital adalah gambar dua dimensi yang bisa ditampilkan pada layar komputer sebagai himpunan/diskrit nilai digital yang disebut piksel. Dalam tinjauan matematis, citra merupakan fungsi kontinu dari intensitas cahaya pada bidang dua dimensi [14]. Foto adalah contoh gambar berdimensi dua yang dapat diolah dengan mudah. Setiap foto dalam bentuk citra digital dapat diolah menggunakan perangkat lunak tertentu. Dalam penelitian ini pengolahan citra yang dilakukan diantaranya grayscale, threshold, segmentasi, resize dan ekstraksi fitur.

2.4 *Grayscale*

Grayscale digunakan untuk mempermudah proses lanjutan dengan cara menyederhanakan nilai-nilai pada matriks citra ke rentang nilai warna puttih dan abu-abu. Berikut proses pengubahan citra ke *grayscale* yang bisa dilihat pada gambar 1 di bawah ini..



Gambar 1 Blok Diagram Proses *Grayscale*

Grayscale adalah suatu proses yang mengubah nilai matriks RGB (*Red*, *Green*, *Blue*) pada citra menuju nilai keabuan dengan menggunakan persamaan 1 di bawah ini.

$$F(x,y) = wR.Red + wG.Green + wB.Blue \quad (1)$$

dimana, x = baris dan y = kolom pada matriks citra. Nilai wR, wG, dan wB masing-masing adalah bobot untuk elemen warna *Red*, *Green* dan *Blue* sesuai pendefisian bobot untuk konversi citra warna ke skala keabuan dimana wR = 0.299, wG = 0.587 dan wB = 0.114 [15].

Cara kerja dari proses ini yaitu:

1) Warna citra dikelompokkan berdasarkan nilai *Red*, *Green* dan *Blue*.
2) Kemudian menggunakan persamaan 1 di atas, maka akan didapatkan nilai warna *grayscale* citra.

3) Nilai *grayscale* yang dihasilkan menggunakan nilai RGB pada setiap piksel.

Berikut adalah contoh gambar hasil grayscale



Gambar II. 2 Hasil *Grayscale*

2.4 Deteksi Tepi *Canny*

Tepi (*edge*) adalah perubahan nilai intensitas derajat keabuan yang cepat atau tiba-tiba (besar) dalam jarak yang singkat. Tujuan mendeteksi tepi sendiri adalah untuk mengelompokkan objek-objek dalam citra, dan juga digunakan untuk menganalisis citra lebih lanjut. Ada banyak algoritma yang digunakan untuk mendeteksi tepi, salah satu diantaranya adalah deteksi tepi *Canny* (*Canny Edge Detection*). *Canny* dikembangkan oleh John F. *Canny* pada tahun 1986 dan menggunakan algoritma multi-tahap untuk mendeteksi berbagai tepi dalam gambar. Langkah Algoritma *Canny* yang berjalan dalam penelitian ini yaitu:

1. *Smoothing* : Mengaburkan gambar untuk menghilangkan noise
2. *Finding gradien* : Tepian harus ditandai pada gambar yang memiliki gradien besar. Pada penelitian ini menggunakan gradian operator *sobel*
3. *Non-maksimum-suppresion* : Hanya nilai maksimum lokal yang harus ditandai sebagai tepi.
4. *Double thresholding* : Tepian yang berpotensi sebagai tepi ditentukan oleh *threshold*ing. Nilai *threshold* tersebut terbagi dua yaitu nilai *threshold* atas dan bawah. Pada penelitian ini nilai *threshold* atas adalah 0.075 dan nilai *threshold* bawah adalah 0.175.

Tidak dapat dipungkiri bahwa semua gambar yang diambil dari kamera atau *scanner* akan berisi sejumlah *noise*. Dalam mencegah noise salah dideteksi sebagai tepian, maka noise harus dikurangi. Oleh karena itu pada langkah pertama citra harus diperhalus dengan mengggunakan *gaussian filter* yang ditunjukkan pada persamaan di bawah ini.

$$g(x,y) = \frac{1}{159}\begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix} \quad (2)$$

Keterangan :

$g(x,y)$ = Matriks kernel konvolusi

Gradien pada setiap piksel gambar yang telah diperhalus ditentukan dengan menerapkan operator *Sobel*. Langkah selanjutnya adalah memperkirakan gradien pada arah x dan y. Hal tersebut ditunjukkan dalam persamaan 3 di bawah ini.

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} dan\ G_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad (3)$$

Magnitudo gradien (juga dikenal sebagai kekuatan tepi) dapat ditentukan sebagai jarak *Euclidean* yang diukur dengan menerapkan hukum *pythagoras* seperti yang ditunjukkan dalam persamaan (4)

$$G[f(x,y)] = \sqrt{G_x^2 + G_y^2} \quad (4)$$

Dimana, G_x dan G_y adalah gradien pada masing-masing arah *x* dan *y* lalu untuk menentukan tepian yang sebenarnya, langkah selanjutnya yaitu menentukan arah tepian menggunakan persamaan 5 berikut.

$$\theta = arctan\left(\frac{|G_y|}{|G_x|}\right) \quad (5)$$

Keterangan :

$\theta = theta\,(sudut\ arah\ tepi)$

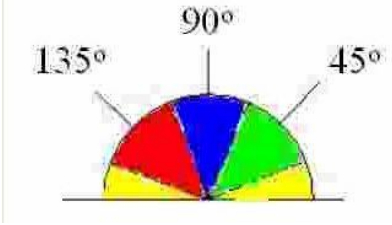
Kemudian jika ditemukan nilai negatif pada matriks setelah melakukan perhitungan melalui persamaan 3 maka digunakan persamaaan 6 berikut untuk mendapatkan nilai positif pada seluruh nilai matriks.

$$M(x,y) = p\ ;\ if\ (M(x,y) < 0)$$
$$untuk\ p = 360 + M(x,y) \quad (6)$$

Keterangan :

$M(x,y) = Matriks\ arah\ tepi$

Setelah seluruh nilai matriks bernilai positif langkah selanjutnya yaitu men*filter* arah tepi menjadi nilai (0, 45, 90 dan 135) dalam derajat. Berikut gambar yang digunakan untuk menentukan rentang arah tepi.



Gambar 3 Arah Tepi *Canny*

Dalam menentukan rentang arah tepi, diambil nilai matriks dari hasil persamaan 5. Dari persamaan tersebut nilai arah tepi yang jatuh dalam *range* kuning (0-22.5 dan 157.5-180) akan dilakukan normalisasi menjadi 0 derajat. Apabila arah tepi yang jatuh dalam *range* hijau (22.5-67.5) maka akan dilakukan normalisasi menjadi 45 derajat. Kemudian arah tepi yang jatuh dalam *range* biru (67.5-112.5) akan dilakukan normalisasi menjadi 90 derajat dan arah tepi yang jatuh dalam *range* merah (112.5-157.5) akan dilakukan normalisasi menjadi 135 derajat.

Pada langkah ini bertujuan untuk mengkorversi tepian yang masih *blurred* pada gambar hasil *magnitude* hingga menghasilkan tepian tajam dengan mempertahankan nilai maksimum lokal dalam gambar gradien dan menghapus segala sesuatu yang lain. Langkah terakhir pada algoritma *canny* adalah menentukan *thresholding* ganda. Kondisi penentuan *thresholding* yaitu gambar/citra sudah ditentukan nilai *non-maksimum suppresion*nya dan mendefinisikan nilai *threshold* atas dan bawah. Nilai *threshold* atas dan bawah tersebut telah didefinisikan dipenjelasan sebelumnya yaitu bernilai 0.075 untuk bagian atas dan 0.175 untuk bagian bawah. Tujuan dari *thresholding* ganda yaitu merubah nilai piksel matriks diantara 0 dan 255. Berikut untuk menentukan *thersholding* ganda ditulis dengan persamaan 5 seperti di bawah ini.

$$Nmax(x,y) = \begin{cases} a & if\ Nmax(x,y) < ta \\ b & if\ Nmax(x,y) > tb \end{cases} \quad (7)$$

dimana,

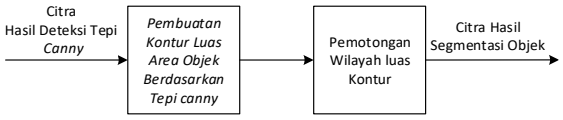
a=0
b=255
$ta = 0.075 \times \max(Nmax(x,y)$
$tb = 0.175 \times \max(Nmax(x,y)$

(5)

Lebih jelasnya nilai *ta* dan tb disebut juga sebagai *hysteresis*. *Hysteresis* adalah penekanan tepian yang tidak terhubung untuk memunculkan tepi-tepi lain yang dideteksi dengan kuat Apabila nilai gradien piksel berada diantara nilai *threshold* atas dan *threshold* bawah, maka piksel tersebut merupakan sebuah tepian lemah.

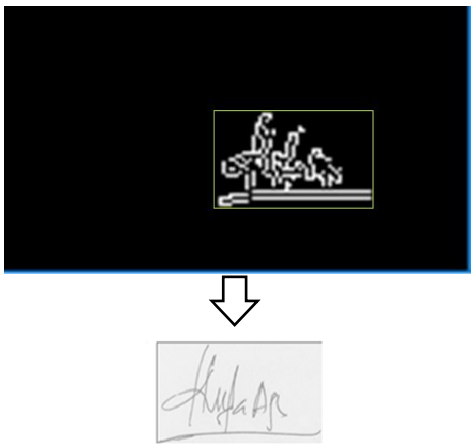
2.6 Segmentasi Objek

Dalam visi komputer, segmentasi adalah proses mempartisi citra digital menjadi beberapa segmen. Tujuan dari segmentasi adalah untuk menyederhanakan atau mengubah penyajian gambar ke sesuatu yang lebih bermakna dan lebih mudah untuk menganalisis[14]. Hasil dari proses segmentasi ini akan digunakan untuk proses tingkat tinggi lebih lanjut yang dapat dilakukan terhadap suatu citra. Berikut ini merupakan proses segmentasi objek pada penelitian ini yang dapat dilihat di gambar 4



Gambar 4 Blok Diagram Proses Segmentasi Objek

Pemotongan objek dilakukan dengan *kontur* yang tersedia di dalam library OpenCV. *Kontur* mengambil nilai deteksi tepi yang dihasilkan operator *canny* untuk menentukan luas wilayah sekeliling objek. Setelah wilayah tersebut didapatkan akan dilakukan pemotongan citra *grayscale* seperti pada gambar 5 berikut ini.



Gambar 5 Citra *Grayscale* Hasil Segmentasi Objek

Hasil Pemotongan citra tersebut mewakili suatu objek pada citra. Pemotongan yang memperoleh objek tersebut dapat memberikan ukuran matriks. yang berbeda-beda apabila dilakukan dengan menggunakan citra input yang lain.

2.7 Resize

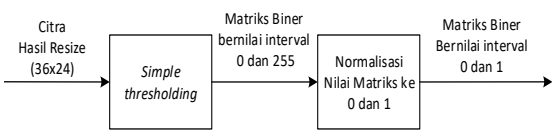
Resize bertujuan untuk menyamakan ukuran citra hasil Segmentasi Objek agar dapat diproses oleh proses selanjutnya yaitu ekstraksi ciri Segmen Vertikal & Horizontal. Dalam penelitan ini digunakan ukuran resize citra 69 x 36 piksel karena ukuran tersebut tidak menurunkan kualitas citra terlalu jauh dari ukuran citra asli, untuk kepentingan analisis pada penelitian ini digunakan ukuran citra sebesar 36 x 24 piksel.Berikut ini adalah gambar 6 hasil proses resize.



Gambar 6 Hasil Proses Resize Citra Masukan

2.8 Binerisasi Citra

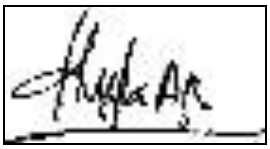
Pada tahap ini matriks hasil proses *resize* yang telah dilakukan Segmentasi Objek disederhanakan lagi nilai matriksnya diantara 0 dan 1 atau dengan kata lain melakukan binerisasi citra. Hal ini dilakukan dengan metode *simple thresholding* serta setelah mendapat hasil matriks *threshold* kemudian membagi semua piksel di koordinat (x, y) dengan 1/255. Berikut adalah blok diagram 7 proses Binerisasi Citra.



Gambar 7 Blok Diagram Proses Binerisasi Citra

Threshold yang digunakan dalam penelitian ini menggunakan *Simple Thresholding* dari library

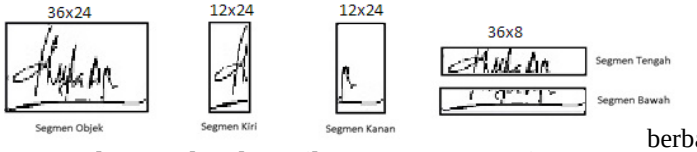
OpenCV yang tujuannya untuk membedakan objek dan *background* dari citra tanda tangan agar ciri informasi lebih mudah untuk dikenali. Berikut matriks hasil proses *thresholding* yang dapat dilihat pada tabel 8 di bawah ini.



Gambar 8 Hasil Threshold

2.9 Segmentasi Vertikal & Horizontal

Pada tahap ini citra hasil proses binerisasi akan dilanjutkan ketahap *segmentasi vertikal dan horizontal*. *Segmentasi vertikal* diproses dengan membagi dua bagian wilayah citra tanda tangan secara vertikal untuk membedakan wilayah fitur *awal kurva* dan *coretan akhir*. Pembagian wilayah ini dilakukan dengan cropping ukuran matriks *threshold* [36x24] ke dua bagian masing-masing dengan ukuran matriks [12x24]. Seperti halnya dengan *segmentasi vertikal* , untuk *segmentasi horizontal* dapat dilakukan dengan membagi dua bagian wilayah citra tanda tangan secara horizontal untuk membedakan wilayah fitur *coretan di tengah* dan *garis bawah*. Dua bagian masing-masing dibagi dengan ukuran matriks [36x8]. Berikut adalah blok diagram 9 proses *segmentasi vertikal dan horizontal*.



Gambar 9 Seluruh Hasil Proses Segmentasi

2.10 Support Vector Machine

Support vector Machine merupakan salah satu metode *machine learning* untuk mengubah data masukan menjadi data vektor. Metode ini dikembangkan oleh Boser, Guyon, Vapnik pada tahun 1992. SVM merupakan suatu teknik untuk menemukan *hyperplane* yang bisa memisahkan dua set data dari dua kelas yang berbeda. *Hyperplane* adalah garis batas pemisah data antar kelas, sedangkan margin adalah jarak antara *hyperplane* dengan data terdekat pada masing-masing kelas. Adapun data terdekat dengan *hyperplane* pada masing-masing kelas inilah yang disebut *support vector*[16].

Label data merupakan anggota dari dua buah kelas data, yaitu +1 dan -1. Data yang tergabung pada kelas -1. Dalam linear *Support vector Machine* pemisah merupakan fungsi linear. Data latih dinyatakan oleh (x_i, y_i) dan x_i = { x_1, x_2, …,x_i} merupakan atribut (fitur) set untuk data latih kelas ke-i. Untuk $y_i \in$ {-1,1} menyatakan label kelas. Pendefinisian persamaan suatu *hyperplane* pemisah dituliskan dengan:

$$wx_i + b = 0 \quad (8)$$

Data x_i yang terbagi ke dalam dua kelas, yang termasuk kelas -1 (sampel negatif) didefinisikan sebagai vektor yang memenuhi pertidaksamaan 9 berikut ini.

$$wx_i + b < 0 \; untuk \; y_i = 1 \quad (9)$$

Sedangkan yang termasuk kelas +1 (sampel positif) memenuhi pertidaksamaan 10 berikut.

$$wx_i + b > 0 \; untuk \; y_i = +1 \quad (10)$$

Dimana:
x_i = data input (vektor)
y_i = label yang diberikan (skalar)
w = nilai dari bidang normal (vektor)
b = bias

Parameter w dan b adalah parameter yang akan dicari nilainya. Bila label data $y_i = -1$, maka pembatas menjadi persamaan 11 berikut ini.

$$wx_i + b \leq -1 \quad (11)$$

Bila label data y_i = +1, maka pembatas menjadi persamaan 10 berikut.

$$wx_i + b \geq +1 \quad (12)$$

Margin terbesar dapat dicari dengan cara memaksimalkan jarak antar bidang pembatas kedua kelas dan titik terdekatnya, yaitu $\frac{2}{|w|}$. Hal ini dirumuskan sebagai permasalahan *quadratic programming* (QP) problem yaitu mencari titik minimal persamaan (13) dengan memperhatikan persamaan (14) berikut.

$$\min \tau(w) = \frac{1}{2}||w||^2 \quad (13)$$

$$y_i(wx_i + b) - 1 \geq 0, (i = 1, \ldots, n) \quad (14)$$

Permasalahan ini ini dapat dipecahkan dengan berbagi teknik komputasi. Lebih mudah diselesaikan dengan mengubah persamaan ke dalam fungsi *Lagrangian*. Nilai a_i adalah *lagrange multiplier* yang bernilai nol atau positif ($a_i \geq 0$). Nilai optimal dari persamaan 14 dapat dihitung dengan meminimalkan L terhadap w, b, dan a. Dapat dilihat pada persamaan 15 sampai 17 berikut ini.

$$\frac{\partial L}{\partial w} = w - \sum_{i=1}^{n} a_i y_i x_i = 0 \quad (15)$$

$$\frac{\partial L}{\partial b} = \sum_{i=1}^{n} a_i y_i = 0 \quad (16)$$

$$\frac{\partial L}{\partial a} = w - \sum_{i=1}^{n} a_i y_i (w^T x_i + b) + \sum_{i=1}^{n} a_i = 0 \quad (17)$$

Maka masalah *Lagrange* untuk klasifikasi dapat dinyatakan pada persamaan 17 berikut dengan memaksimalkan L terhadap a_i, persamannya menjadi persamaan 18 berikut ini.

$$Max \sum_{i=1}^{n} a_i - \frac{1}{2} \sum_{i=1,j=1}^{n} a_i y_i y_j{}^T x_i x_j{}^T \quad (18)$$

Dengan memperhatikan persamaan 19 berikut ini.

$$\sum_{i=1}^{n} a_i y_i = 0, a_i > 0 \; (i, j = 1, \ldots, n) \quad (19)$$

Mencari nilai x_i dan y_i dapat dilakukan ketika sudah didapatkan hasil proses ekstraksi ciri dan inisialisasi kelas . Hasil tersebut diubah ke dalam

bentuk format data svm (*x*), sedangkan data kelas menjadi label data svm (*y*).

Nilai a_i didapatkan dengan cara adalah mengubah setiap data masukan menjadi nilai vektor (*support vector*) = $\begin{pmatrix} x_i \\ y_i \end{pmatrix}$. Kemudian nilai vektor dari setiap data masukan dimasukkan ke persamaan 20 kernel *trick phi* (φ) berikut ini.

$$\varphi \begin{bmatrix} x_i \\ y_i \end{bmatrix} = \begin{cases} \sqrt{x_n^2 + y_n^2} > 2, maka \begin{bmatrix} \sqrt{x_n^2 + y_n^2} - x + |x - y| \\ \sqrt{x_n^2 + y_n^2} - y + |x - y| \end{bmatrix} \\ \sqrt{x_n^2 + y_n^2} < 2, maka \begin{bmatrix} x_i \\ x_i \end{bmatrix} \end{cases} \quad (20)$$

Nilai *x* didapatkan dari persamaan 21 kernel linear untuk x berikut.

$$\sum_{i=1,j=1}^{n} x_i^T x_j \; (i, j = 1, \dots, n) \quad (21)$$

Nilai *y* didapatkan dari persamaan 22 kernel linear untuk y berikut.

$$\sum_{i=1,j=1}^{n} y_i^T y_j \; (i, j = 1, \dots, n) \quad (22)$$

Jarak tegak lurus yang optimal dapat dipertimbangkan dengan mempertimbangkan vektor positif, maka hasil perhitungan dari substitusi nilai *x* dan nilai *y* ke persamaan 22 diberi nilai bias = 1 . Kemudian cari parameter a_i, dengan terlebih dahulu mencari nilai fungsi setiap abstrak menggunakan persamaan 20, lalu mencari nilai a_i pada persamaan linear menggunakan persamaan 23 dengan memperhatikan $i, j = 1, \dots, n$ berikut.

$$\sum_{i=1,j=1}^{n} a_i T_i^T T_j \quad (22)$$

$$\sum_{i=1,j=1}^{n} a_i T_i^T T_j = y_i \quad (23)$$

Dimana, T merupakan nilai *support vector.*

Setelah parameter a_i didapatkan, kemudian masukkan ke persamaan 24 berikut.

$$\tilde{W} = \sum_{i=1}^{n} a_i T_i \quad (24)$$

Hasil yang didapatkan menggunakan persamaan 24, selanjutnya digunakan persamaan 24 untuk mendapatkan nilai w dan b yaitu sebagai berikut.

$$y = wx + b \quad (25)$$

Sedemikian sehingga didapatkanlah nilai w dan nilai b atau nilai *hyperplane* untuk mengklasifikasikan kedua kelas.

Berikut ini fungsi *kernel* yang dapat mengubah data set *non-linear* menjadi *linier* dalam *space* baru.

a) Linier

$$K(x_i x_j) = x^T x \quad (26)$$

b) Polynomial

$$K(x_i x_j) = (x^T x_i + 1)^p \quad (27)$$

c) Gaussian

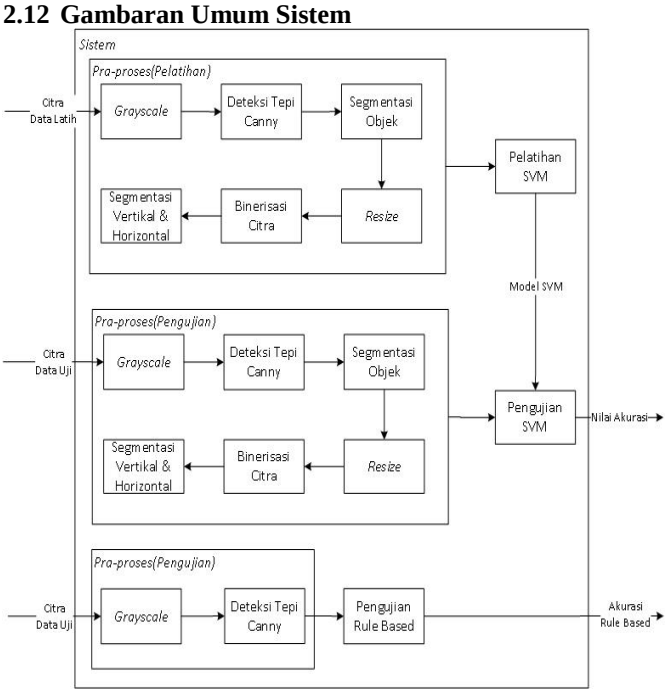
$$K(x_i x_j) = \exp - \|x_1 - x_2\|^2 / 2\sigma \quad (28)$$

Pada penelitian ini, kernel yang digunakan ialah kernel linier dan proses klasifikasi *Support Vector Machine* menggunakan LIBSVM Scikit-learn python.

2.11 Rule Based

Metode *rule based* adalah salah satu metode dalam sistem pendukung keputusan yang di dalamnya memiliki basis pengetahuan. Mekanisme dari *rule based* dilakukan dengan cara menguji aturan-aturan satu persatu dalam urutan tertentu. Saat tiap aturan diuji *rule based* akan mengevaluasi apakah kondisinya benar atau salah. Jika kondisinya benar maka kemudian dilanjutkan ke aturan berikutnya diuji. Sebaliknya kondisinya salah, aturan itu tidak disimpan dan aturan berikutnya diuji. Proses ini akan berulang sampai seluruh basis aturan teruji dengan berbagai kondisi.

2.12 Gambaran Umum Sistem



Gambar 11 arsitektur umum sistem

Pada blok diagram di atas, proses awal yang dilakukan yaitu memasukkan citra tanda tangan ke dalam sistem. Langkah selanjutnya melalui tahap pengolahan citra meliputi *grayscale*, *deteksi tepi canny*, *segmentasi objek*, *resize*, *binerisasi citra* dan ekstraksi ciri dengan *segmentasi vertikal & horizontal*. Hasil dari pengolahan citra akan didapatkan nilai matriks dari setiap citra masukan, kemudian diolah kembali pada tahap klasifikasi dengan metode SVM baik pelatihan maupun pengujian. Demikian pula dengan *rule based* akan tetapi tidak melalui tahap pelatihan. Hasil dari tahap pengujian yaitu berupa akurasi klasifikasi pengenalan pola pada tanda tangan.

2.12 Citra Tanda Tangan

Data masukan yang akan diolah *Support vector Machine* adalah hasil *scan* gambar tanda tangan yang tersimpan dalam bentuk file .jpg. Dalam analisis ini jumlah data yang diambil dari koresponden berjumlah 100 data tanda tangan yang berbeda terdiri dari masing-masing 50 data tanda tangan untuk tiap fiturnya. Berikut data yang contoh data yang telah dikumpulkan dari koresponden seperti gambar 12 di bawah ini.



GAMBAR 12 CITRA MASUKAN

Gambar 12 di atas memiliki ukuran sebesar 1000x800 piksel. Selanjutnya sebagai contoh, akan digunakan gambar (b) untuk citra masukan sistem seperti pada gambar 3.3 di bawah ini.

2.13 Pengolahan Citra Tanda Tangan

Dalam pengolahan citra dilakukan beberapa proses agar citra input dapat digunakan pada tahap klasifikasi menggunakan metode *Support Vector Machine* (SVM). Berikut adalah penjabaran dari proses pengolahan citra yang dilakukan sesuai dengan arsitektur sistem yang dibuat.

1. Grayscale

Proses *grayscale* citra. *Grayscale* merupakan tahap awal yang dilakukan dalam tahap pengolahan citra. Hasil dari proses ini nantinya menghasilkan citra dengan range warna dari 0 sampai dengan 255 yang merupakan skala keabuan.

2. Deteksi tepi *canny*

Hasil akhir dari proses ini adalah citra berlatar hitam yang memiliki nilai tepi yang dideteksi sebagai objek pada citra.

3. Segmentasi

Dalam proses segmentasi, citra harus sudah melalui proses deteksi tepi. Hal ini dilakukan agar sistem mampu memotong objek tanda tangan yang berbeda ukuran.

4. Binarisasi Citra

Langkah selanjutnya dalam pengolahan citra adalah thresholding. Proses ini dilakukan terhadap citra *grayscale*. Metode *threshold* yang digunakan dalam penelitian ini adalah *simple threshold*. Hasil akhir dari proses ini adalah citra hitam putih.

5. Resize

Citra hasil segmentasi, tentu ukurannya masih beragam. Oleh karena itu perlu diseragamkan menjadi suatu ukuran tertentu agar sistem dapat konsisten membaca citra. Dalam penelitian ini citra di-*resize* menjadi ukuran yang sudah ditentukan pada masing-masing fitur.

6. Ekstraksi Ciri (segmentasi vertikal&horizontal)

Hasil dari proses ini yaitu berupa deretan angka biner pada tiap wilayah fitur yang nantinya digunakan sebagai data input pada tahap klasifikasi menggunakan metode *Support Vector Machine* (SVM).

2.14 Pengujian Akurasi

Pada pengujian akurasi akan dihitung nilai ketepatan atau kecocokan dari hasil aplikasi pengenalan pola tanda tangan dalam mendeteksi kepribadian. Dalam penelitian ini pengujian parameter pelatihan yang digunakan sama dan akurasi yang dihasilkan pengujian svm serta ruled yaitu sbb.

Jenis Pola Tanda Tangan	**Jumlah Data Uji**	**Hasil Klasifikasi**		**Akurasi SVM**
		Benar	**Salah**	
Cangkang	50	21	29	61%
Kurva Awal	50	27	23	55%
Coretan Akhir	50	25	25	50%
Coretan Tengah	50	45	5	89,5%
Garis Bawah	50	34	16	67,3%
Rata-rata Akurasi				60,7%

Jenis Pola Tanda Tangan	**Jumlah Data Uji**	**Hasil Klasifikasi**		**Akurasi Rulebased**
		Benar	**Salah**	
Margin Ekstrim (*x*)	50	35	15	70%
Margin Ekstrim (*y*)	50	30	20	60%
Struktur titik	50	22	28	44%
Garis Akhir Terputus	50	17	33	34%
Struktur Terpisah	50	28	22	56%
Rata-rata Akurasi				52,8%

3. PENUTUP

3.1 Kesimpulan

Berdasarkan hasil penelitian dengan 50 data latih dan 50 data uji berupa citra tanda tangan menghasilkan akurasi SVM paling tinggi sebesar 89% dan paling rendah sebesar 41% dengan rata-rata akurasi sebesar 60,7%. Pengujian menggunakan metode *rulebased* dengan 50 data uji menghasilkan akurasi paling tinggi sebesar 70% dan paling rendah 34% dengan rata-rata akurasi sebesar 52,8%. Berikut ini adalah penjelasan dari peyebab nilai akurasi yang dihasilkan sistem tidak begitu baik akurasinya.

1. Data latih yang diambil polanya diseleksi keseragamannya sama sedangkan data yang diuji yang diambil terlalu beragam atau variatif menjadikan nilai akurasi rendah.

2. Ekstraksi ciri yang digunakan untuk mengekstrak informasi masih belum baik dalam mewakili informasi sebagai pola dari tanda tangan tersebut sehingga dapat menurunkan kualitas klasifikasi.

Sehingga dapat diambil kesimpulan bahwa metode SVM sebenarnya cocok untuk digunakan dalam sistem pendeteksi kepribadian berdasarkan pola tanda tangan, akan tetapi dengan meningkatkan proses ekstraksi ciri yang baik dan menambahkan data latih yang semakin beragam polanya agar dapat mengklasifikasikan dengan lebih baik serta menambahkan fitur tanda tangan yang belum ada pada sistem.

3.2 Saran

Berdasarkan hasil pengujian yang telah dilakukan, maka saran yang diharapkan adalah sebagai berikut.

1. Menambahkan atau mengganti proses ekstraksi ciri seperti metode PCA(*principal compenent analysit)* untuk mereduksi matriks dan tetap menjaga informasi pola citra yang diambil dengan baik atau mengurangi informasi ciri yang tidak diperlukan.
2. Menggunakan kernel yang berbeda pada metode SVM sebagai bahan perbandingan seperti kernel RBF, polynomial dsb.
3. Menambahkan fitur yang belum ada pada penelitian ini untuk melengkapi sistem pendeteksi kepribadian berdasarkan pola tanda tangan.
4. Menambahkan data latih pola yang lebih variatif agar dapat mendeteksi berbagai jenis pola tanda tangan.

DAFTAR PUSTAKA

[1] Abhicandra, *Panduan Lengkap Grafologi*. Yogyakarta: Araska, 2017.

[2] E. C. Djamal, R. Darmawati, and S. N. Ramdlan, "Application image processing to predict personality based on structure of handwriting and signature," *Proceeding - 2013 Int. Conf. Comput. Control. Informatics Its Appl. "Recent Challenges Comput. Control Informatics", IC3INA 2013*, pp. 163–168, 2013.

[3] K. N. Lakshmi, A. Keerthana, and P. R. Lakshmi, "Handwriting Analysis Based Human Personality Prediction," vol. 8, no. 5, pp. 105–110, 2017.

[4] S. Prasad, ivek K. Singh, and A. Sapre "Handwriting Analysis based on Segmentation Method for Prediction of Human Personality using Support Vector Machine," *Int. J. Comput. Appl.*, vol. 8, no. 12, pp. 25–29, 2010.

[5] R. Kumar, P. Singhal, and E. C. E. Vlsi, "Review on Offline Signature Verification by SVM," *Int. Res. J. Eng. Technol.*, vol. 4, no. 6, pp. 1771–1773, 2017.

[6] V. Iranmanesh *et al.*, "Online handwritten signature verification using neural network classifier based on principal component analysis.," *ScientificWorldJournal.*, vol. 2014, p. 381469, 2014.

[7] J. Hu, Z. Guo, Z. Fan, and Y. Chen, *Offline signature verification using local features and decision trees*, vol. 31, no. 3. 2017.

[8] E. C. Djamal, S. N. Ramdlan, J. Informatika, T. Jenderal, and S. Cimahi, "Pengenalan Pola Tanda Tangan Menggunakan MLP Dalam Identifikasi Kepribadian," pp. 2–4, 2013.

[9] Thoyib, "Pengenalan Pola Tanda Tangan Menggunakan Learning Vector Quantization Dalam Identifikasi Kepribadian," Perpustakaan UNIKOM, 2015.

[10] D. Putra, *Pengolahan Citra Digital*. Yogyakarta: Andi, 2010.

[11] S. R. Sternberg, "Grayscale Morphology.," *Comput. vision, Graph. image Process.*, vol. 35, no. 3, pp. 333–355, 1986.