

ANALISIS *PEN PRESSURE* TULISAN TANGAN UNTUK MENGIDENTIFIKASI KEPRIBADIAN SESEORANG MENGGUNAKAN *SUPPORT VECTOR MACHINE (SVM)*

Ganis Khufad Arridho¹, Sukmawati Nur Endah, S.Si, M.Kom², Aris Sugiharto, S.Si, M.Kom²

Ilmu Komputer / Informatika FSM Universitas Diponegoro

ganis.khufad@gmail.com, sukma_ne@undip.ac.id, aris.sugiharto@undip.ac.id

Abstrak

Sifat kepribadian seseorang dapat tergambar dari tulisan tangannya. Ilmu yang mempelajari kepribadian seseorang lewat tulisan tangannya dikenal dengan istilah grafologi. Salah satu cara mengidentifikasi tulisan tangan adalah dengan menganalisis pola tekanan tulisan tangan atau yang disebut pola *pen pressure*. Pola tekanan tulisan tangan memiliki dua klasifikasi yaitu tekanan tulisan tebal dan tekanan tulisan tipis. Sistem yang dibangun pada tugas akhir ini adalah analisis *pen pressure* tulisan tangan untuk mengidentifikasi kepribadian seseorang. Terdapat tiga tahap utama untuk menyelesaikan permasalahan yaitu *preprocessing*, ekstraksi fitur, dan klasifikasi. Metode *preprocessing* yang digunakan adalah *grayscale* citra dan binerisasi citra. Penjumlahan *pixel* hitam digunakan untuk proses ekstraksi fitur. Untuk pengklasifikasian digunakan metode *Support Vector Machine (SVM)*. Pada tahap pengujian digunakan sebanyak 70 sampel data tulisan tangan, 35 sampel digunakan untuk training dan 35 lainnya digunakan untuk testing. Hasil pengujian dengan melakukan 5 kali eksperimen menunjukkan rata-rata akurasi sebesar 96,54%.

Kata kunci : Grafologi, *Pen pressure*, *Support Vector Machine*

Abstract

The nature of an individual's personality can be reflected from his or her handwriting. The study of an individual's personality by analyzing his or her handwriting is known as graphology. The patterns of the handwriting pressure are classified into two; heavy pen pressure and light pen pressure. The system built in this thesis is the analysis of the handwriting pen pressure to identify one's character. There were three steps which were employed to solve the problem; preprocessing, feature extraction, and classification. The preprocessing methods which were conducted were image grayscale and image binarization. The calculation of the black pixels was done for the feature extraction process. Meanwhile, the method which was used for the classification was Support Vector Machine (SVM). In the stages of testing, 70 data samples of handwriting were used, 35 samples were used for training and the other 35 were used for testing. The test results by performing 5 times experiments show 96.54% average accuracy.

Keywords : Graphology, Pen pressure, Support Vector Machine

1. Pendahuluan

Manusia memiliki karakteristik dan sifat yang bermacam-macam. Salah satu cara untuk mengetahui karakteristik tersebut adalah melalui tulisan tangannya. Tulisan tangan seseorang sama halnya seperti DNA atau sidik jari seseorang yang memiliki keunikan

tersendiri untuk mencirikan karakteristik seseorang. Ilmu untuk mengetahui kepribadian seseorang melalui tulisan tangan dikenal dengan istilah grafologi.

Grafologi sudah banyak diteliti, salah satunya adalah Wilhiem Prayer. Dalam sebuah jurnal : Deutche Graphologische Gesellschaft

1) Mahasiswa Jurusan Teknik Informatika FSM UNDIP

2) Staf Pengajar Jurusan Teknik Informatika FSM UNDIP

yang terbit tahun 1895 Wilhiem Prayer mengemukakan bahwa tulisan tangan pada dasarnya adalah tulisan otak [4]. Begitu juga grafologi dibahas pada jurnal internasional yang dibuat oleh Vivek Kumar Singh dkk.

Dalam jurnal tersebut, secara garis besar, untuk mengetahui karakteristik seseorang dapat dilihat dari beberapa ciri diantaranya adalah ukuran huruf, kemiringan kata, penekanan tulisan (*pen pressure*), jarak spasi antar huruf, dan jarak spasi antar kata. Pola *pen pressure* memiliki dua jenis tipe tulisan yaitu *heavy pen pressure* (tulisan tebal) dan *light pen pressure* (tulisan tipis). Selain kepribadian, pola *pen pressure* dapat menunjukkan bagaimana keadaan emosional seseorang pada saat menulis.

Metode untuk membangun sistem pengenalan tulisan tangan ini diantaranya adalah *Artificial Neural Network (ANN)*, Algoritma Genetika, *Support Vector Machine (SVM)*, dan *Optical Character Recognition (OCR)*. *Support Vector Machine (SVM)* pertama kali diperkenalkan oleh Vapnik pada tahun 1992 sebagai rangkaian harmonis konsep-konsep unggulan dalam bidang *pattern recognition*. Sebagai salah satu metode *pattern recognition*, usia SVM terbilang masih relatif muda. Walaupun demikian, evaluasi kemampuannya dalam berbagai aplikasinya menempatkannya sebagai *state of the art* dalam *pattern recognition*, dan dewasa ini merupakan salah satu tema yang berkembang dengan pesat. SVM adalah metode *learning machine* yang bekerja atas prinsip *Structural Risk Minimization (SRM)* dengan tujuan menemukan *hyperplane* terbaik yang memisahkan dua buah *class* pada *input space*.

2. Dasar Teori

2.1. Grafologi

Setiap tulisan yang dibuat oleh manusia mencerminkan kepribadian orang tersebut. Teknik untuk mengetahui kepribadian seseorang melalui tulisan tangannya dikenal dengan istilah Grafologi. Grafologi adalah ilmu yang mempelajari karakter seseorang dengan cara menganalisa tulisan tangannya, buku pertama tentang grafologi ditulis oleh

Camillo Baldi seorang dokter asal Italia pada tahun 1622. Tahun 1872, Jean Michon menerbitkan bukunya yang menjadi buku pokok grafologi pada saat itu. Tak lama kemudian, universitas di Eropa mulai memberi gelar Ph.D. atau Master dibidang ini [3].

2.2. Pola *Pen Pressure* Tekanan Tangan

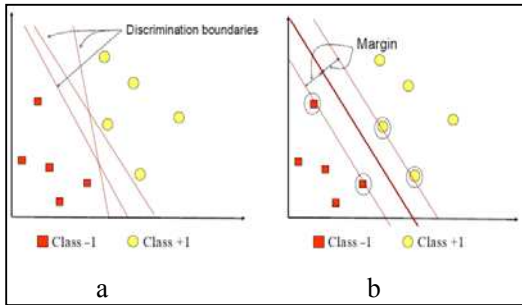
Tekanan tulisan tangan (*Pen Pressure*) bercerita banyak tentang bagaimana seseorang mengekspresikan perasaannya. Terdapat dua kecenderungan tekanan umum pada penulisan yaitu tekanan tebal (*Heavy Pen Pressure*), dan tekanan tipis (*Light Pen Pressure*). Cara mudah melihat seorang menulis dengan tekanan tebal atau tipis adalah dengan memeriksa bagian belakang kertas. Jika seseorang menulis dengan tekanan tebal maka bagian belakang kertas akan tampak menonjol alias permukaan kertas menjadi tidak mulus. Sebaliknya, jika seseorang menulis dengan tekanan tipis maka bagian belakang kertas akan datar dan mulus [6].

2.3. *Preprocessing* Citra Digital

Preprocessing adalah tahap pertama yang harus dilakukan pada proses pengenalan pola tulisan tangan. Tahap ini sangat penting untuk menentukan keberhasilan suatu proses pengenalan pola tulisan tangan. Beberapa proses yang dapat dilakukan pada tahap *preprocessing* antara lain proses *grayscale* dan binerisasi citra.

2.4. *Support Vector Machine (SVM)*

SVM berada dalam satu kelas dengan ANN (*Artificial Neural Network*) dalam hal fungsi dan kondisi permasalahan yang dapat diselesaikan. Keduanya masuk dalam kelas *supervised learning*. Baik para ilmuwan maupun praktisi telah banyak menerapkan teknik ini dalam menyelesaikan masalah-masalah nyata dalam kehidupan sehari-hari. Baik dalam masalah *gene expression, analysis*, finansial, cuaca hingga di bidang kedokteran. Terbukti dalam banyak implementasi, SVM memberi hasil yang lebih baik dari ANN, terutama dalam hal solusi yang dicapai.



Gambar 2.1. Usaha SVM dalam menemukan *hyperplane* terbaik

2.4.1 Pengenalan Pola Menggunakan Support Vector Machine

Konsep SVM dapat dijelaskan secara sederhana sebagai usaha mencari *hyperplane-hyperplane* terbaik yang berfungsi sebagai pemisah dua buah *class* pada *input space*. Gambar 2.6.a. memperlihatkan beberapa pola yang merupakan anggota dari dua buah *class*: -1 dan +1. Pola yang tergabung pada *class* -1 disimbolkan dengan warna merah (kotak), sedangkan pola pada *class* +1, disimbolkan dengan warna kuning (lingkaran). Problem klasifikasi dapat diterjemahkan dengan usaha menemukan garis (*hyperplane*) yang memisahkan antara kedua kelompok tersebut. Berbagai alternatif garis pemisah (*discrimination boundaries*) dapat dilihat pada Gambar 2.1.a.

Hyperplane pemisah terbaik antara kedua *class* dapat ditemukan dengan mengukur *margin hyperplane* tersebut dan mencari titik maksimalnya. *Margin* adalah jarak antara *hyperplane* tersebut dengan *pattern* terdekat dari masing-masing *class*. *Pattern* yang paling dekat ini disebut sebagai *support vector*. Garis solid pada Gambar 2.1.b menunjukkan *hyperplane* yang terbaik, yaitu yang terletak tepat pada tengah-tengah kedua *class*, sedangkan titik merah dan kuning yang berada dalam lingkaran hitam adalah *support vector*. Usaha untuk mencari lokasi *hyperplane* ini merupakan inti dari proses pembelajaran pada SVM.

Data yang tersedia dinotasikan sebagai $x_i \in R^d$ sedangkan label masing-masing dinotasikan $y_i \in \{-1, +1\}$ untuk $i =$

$1, 2, \dots, l$, yang mana l adalah banyaknya data. Diasumsikan kedua *class* -1 dan +1 dapat terpisah secara sempurna oleh *hyperplane* berdimensi d , yang didefinisikan oleh persamaan 1.

$$\bar{w} \cdot \bar{x} + b = 0 \quad (1)$$

Pattern \bar{x}_i yang termasuk *class* -1 (sampel negatif) dapat dirumuskan sebagai *pattern* yang memenuhi pertidaksamaan sebagaimana dapat dilihat pada persamaan 2.

$$\bar{w} \cdot \bar{x}_i + b \leq -1 \quad (2)$$

Sedangkan *pattern* \bar{x}_i yang termasuk *class* +1 (sampel positif) dapat dilihat pada persamaan 3.

$$\bar{w} \cdot \bar{x}_i + b \geq +1 \quad (3)$$

\bar{w} adalah normal bidang dan b adalah posisi bidang relatif terhadap pusat koordinat. *Margin* terbesar dapat ditemukan dengan memaksimalkan nilai jarak antara *hyperplane* dan titik terdekatnya, yaitu $1/\|\bar{w}\|$. Hal ini dapat dirumuskan sebagai *Quadratic Programming (QP) problem*, yaitu mencari titik minimal persamaan 4, dengan memperhatikan *constraint* persamaan 5.

$$\begin{aligned} \min_{\bar{w}} \quad & \frac{1}{2} \|\bar{w}\|^2 \quad (4) \\ \text{s.t.} \quad & y_i (\bar{x}_i \cdot \bar{w} + b) - 1 \geq 0, \quad \forall_i \quad (5) \end{aligned}$$

Problem ini dapat dipecahkan dengan berbagai teknik komputasi, diantaranya *Lagrange Multiplier*.

$$\begin{aligned} L(\bar{w}, b, \alpha) = & \\ \frac{1}{2} \|\bar{w}\|^2 - \sum_{i=1}^l \alpha_i (y_i (\bar{x}_i \cdot \bar{w} + b) - 1) & \\ (i = 1, 2, \dots, l) & \quad (6) \end{aligned}$$

α_i adalah *Lagrange Multiplier*, yang bernilai nol atau positif ($\alpha_i \geq 0$). Nilai optimal dari persamaan 6 dapat dihitung dengan meminimalkan L terhadap \bar{w} dan b , dan memaksimalkan L terhadap α_i . Dengan

memperhatikan sifat bahwa pada titik optimal gradient $L=0$, persamaan 6 dapat dimodifikasi sebagai maksimalisasi *problem* yang hanya mengandung α_i saja, sebagaimana dapat dilihat pada persamaan 7.

Maximize:

$$\sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i,j=1}^l \alpha_i \alpha_j \gamma_{ij} \bar{x}_i \cdot \bar{x}_j \quad (7)$$

dengan:

$$\alpha_i \geq 0 \quad (i = 1, 2, \dots, l) \quad \sum_{i=1}^l \alpha_i \gamma_{ij} = 0 \quad (8)$$

Dari hasil perhitungan ini diperoleh α_i yang kebanyakan bernilai positif. Data yang berkorelasi dengan α_i yang positif inilah yang disebut sebagai *support vector*.

2.4.2 Metode Kernel Support Vector Machine

Terdapat banyak teknik *machine learning* yang dikembangkan dengan asumsi kelinieran sehingga algoritma yang dihasilkan terbatas untuk kasus-kasus yang linier. Oleh karena itu, jika suatu kasus klasifikasi memperlihatkan ketidaklinieran, algoritma seperti perceptron tidak dapat mengatasinya [12].

Pada umumnya masalah dalam domain dunia nyata (*real world problem*) jarang yang bersifat *linear separable*. Kebanyakan bersifat *non linear*. Untuk menyelesaikan problem *non linear*, SVM dimodifikasi dengan memasukkan fungsi *kernel*.

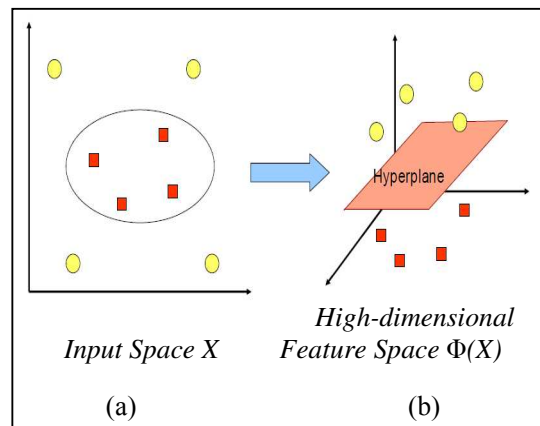
Dalam *non linear SVM*, pertama-tama data \bar{x} dipetakan oleh fungsi $\Phi(\bar{x})$ ke ruang vektor yang berdimensi lebih tinggi. Pada ruang vektor yang baru ini, *hyperplane* yang memisahkan kedua *class* tersebut dapat dikonstruksikan. Hal ini sejalan dengan teori Cover yang menyatakan bahwa:

“Jika suatu transformasi bersifat *non linear* dan dimensi dari *feature space* cukup tinggi, maka data pada *input space* dapat dipetakan ke *feature space* yang baru, dimana *pattern-pattern* tersebut pada *probabilitas* tinggi dapat dipisahkan secara *linear*” [9].

Ilustrasi dari konsep tersebut dapat dilihat pada Gambar 2.2. Pada Gambar 2.2.a

diperlihatkan data pada *class* warna kuning dan data pada *class* merah yang berada pada *input space* berdimensi dua tidak dapat dipisahkan secara *linear*. Selanjutnya Gambar 2.2.b menunjukkan bahwa fungsi Φ memetakan tiap data pada *input space* tersebut ke ruang vektor baru yang berdimensi lebih tinggi (dimensi 3), kedua *class* dapat dipisahkan secara *linear* oleh sebuah *hyperplane*. Notasi matematika dari *mapping* ini adalah sebagai berikut:

$$\Phi: K^d \rightarrow K^q \quad d < q \quad (9)$$



Gambar 2.2. Fungsi Φ memetakan data ke ruang vektor yang berdimensi lebih tinggi

Tabel 2.1. Kernel yang umum dipakai dalam SVM

Jenis Kernel	Definisi
Linear	$K(\bar{x}_i, \bar{x}_j) = \bar{x}_i \cdot \bar{x}_j$
Polynomial l	$K(\bar{x}_i, \bar{x}_j) = (\bar{x}_i \cdot \bar{x}_j + 1)^l$
Gaussian atau Radial Basis Function (RBF)	$K(\bar{x}_i, \bar{x}_j) = \exp\left(-\frac{\ \bar{x}_i - \bar{x}_j\ ^2}{2\sigma^2}\right)$
Tangent Hyperbolic (Sigmoid)	$K(\bar{x}_i, \bar{x}_j) = \tanh(c\bar{x}_i \cdot \bar{x}_j + d)$

Tabel 2.1. merupakan contoh jenis kernel

yang biasanya dipakai dalam SVM [9][12]. Pemetaan ini dilakukan dengan menjaga topologi data, dalam artian dua data yang berjarak dekat pada *input space* akan berjarak dekat juga pada *feature space*, sebaliknya dua data yang berjarak jauh pada *input space* akan juga berjarak jauh pada *feature space*. Selanjutnya, proses pembelajaran pada SVM dalam menemukan titik-titik *support vector*, hanya bergantung pada *dot product* dari data yang sudah ditransformasikan pada ruang baru yang berdimensi lebih tinggi, yaitu $\Phi(\vec{x}_1), \Phi(\vec{x}_2)$.

Pada umumnya, transformasi Φ ini tidak diketahui, dan sangat sulit untuk difahami secara mudah, oleh karena itu perhitungan *dot product* tersebut sesuai teori Mercer dapat digantikan dengan fungsi kernel $K(\vec{x}_1, \vec{x}_2)$. Hal ini disebut sebagai *Kernel Trick*, yang dirumuskan sebagai berikut:

$$K(\vec{x}_1, \vec{x}_2) = \Phi(\vec{x}_1) \cdot \Phi(\vec{x}_2). \quad (10)$$

Kernel trick memberikan berbagai kemudahan, karena dalam proses pembelajaran SVM, untuk menentukan *support vector* (SV), hanya cukup mengetahui fungsi kernel yang dipakai, dan tidak perlu mengetahui wujud dari fungsi *non linear* Φ . Berbagai jenis fungsi kernel dikenal, sebagaimana dirangkumkan pada Tabel 2.1. Selanjutnya hasil klasifikasi dari data \vec{x} diperoleh dari persamaan berikut:

$$f(\Phi(\vec{x})) = \vec{w} \cdot \Phi(\vec{x}) + b \quad (11)$$

$$= \sum_{i=1}^n \alpha_i y_i \Phi(\vec{x}) \cdot \Phi(\vec{x}_i) + b \quad (12)$$

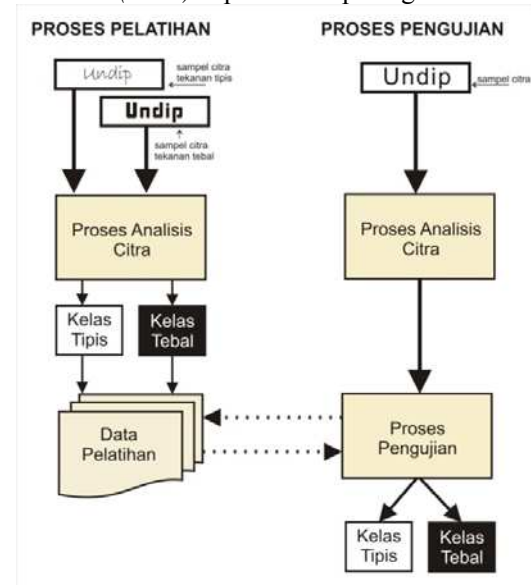
$$= \sum_{i=1}^n \alpha_i y_i K(\vec{x}, \vec{x}_i) + b \quad (13)$$

SV (*Support Vector*) pada persamaan 12 dan 13 dimaksudkan dengan *subset* dari *training set* yang terpilih sebagai *support vector*, dengan kata lain data \vec{x}_i yang berkorespondensi pada $\alpha_i \geq 0$.

3. Analisis dan Perancangan Sistem

3.1. Gambaran Umum Sistem

Secara umum proses analisis tekanan tulisan tangan menggunakan *Support Vector Machine* (SVM) dapat dilihat pada gambar 3.1.



Gambar 3.1 Gambaran Umum

Terdapat dua proses utama dalam mengerjakan aplikasi ini yaitu proses pelatihan dan proses pengujian. Proses pelatihan dibangun dengan tujuan untuk mendapatkan data pelatihan yang akan digunakan pada proses pengujian.

Pada proses pelatihan citra yang masuk akan dianalisis, keluaran dari proses analisis ini berupa data jumlah *pixel* hitam dari citra masukannya. Selanjutnya, jika jumlah *pixel* hitam kurang dari batas ambang maka citra akan digolongkan kedalam kelas tipis, sedangkan jika jumlah *pixel* hitam melebihi batas ambang maka akan digolongkan kedalam kelas tebal.

3.2. Analisis Kebutuhan

Tabel 3.1 SRS Aplikasi Analisis Tulisan Tangan

SRS ID	Deskripsi
SRS-ATT-F01	Menerima masukkan <i>file</i> citra digital dengan format <i>bitmap</i>

SRS-ATT-F02	Menampilkan informasi citra digital
SRS-ATT-F03	Melakukan proses <i>preprocessing</i> citra digital
SRS-ATT-F04	Melakukan proses Ekstraksi Fitur Citra
SRS-ATT-F05	Menghitung lama waktu proses pengujian
SRS-ATT-F06	Menampilkan lama waktu proses pengujian
SRS-ATT-F07	Menghitung jumlah <i>pixel</i> hitam
SRS-ATT-F08	Menampilkan jumlah <i>pixel</i> hitam
SRS-ATT-F09	Melakukan proses pengujian
SRS-ATT-F10	Menampilkan hasil proses pengujian

3.3. Pemodelan Fungsional

3.3.1 Data Context Diagram



Gambar 3.2. DCD aplikasi Analisis Tulisan Tangan

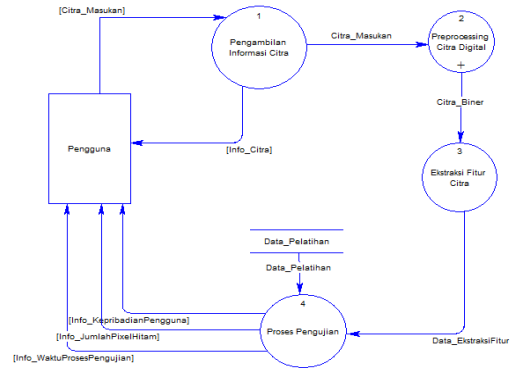
Gambar 3.2. menjelaskan bahwa pengguna dapat memberikan masukan kepada sistem berupa *file* citra digital dalam format bitmap yang kemudian dilakukan proses pengenalan pola tekanan tulisan tulisan tangan. Keluaran dari aplikasi ini adalah informasi citra berupa nama file, ukuran, panjang, lebar, jumlah titik hitam, informasi waktu proses pengujian dan informasi kepribadian.

3.3.2 Data Flow Diagram

Sebagaimana DCD yang ditunjukkan pada gambar 3.2., maka dapat dibuat diagram alir data atau *Data Flow Diagram (DFD)* yang merupakan penjelasan lebih rinci dari DCD pada setiap tahapan

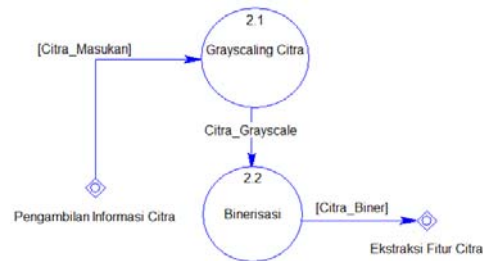
a. DFD Level 1

Dari DCD pada gambar 3.2., maka dapat dibuat DFD yang lebih rinci lagi, seperti pada gambar 3.3.



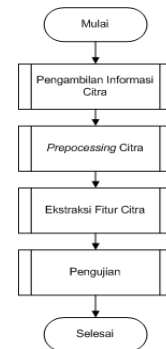
Gambar 3.3. DFD Level 1 aplikasi Analisis Tulisan Tangan

b. DFD Level 2



Gambar 3.4. DFD Level 2 aplikasi Analisis Tulisan Tangan

3.4. Perancangan Proses Analisis Tulisan Tangan



Gambar 3.5. *Global flowchart* proses Analisis Tulisan Tangan

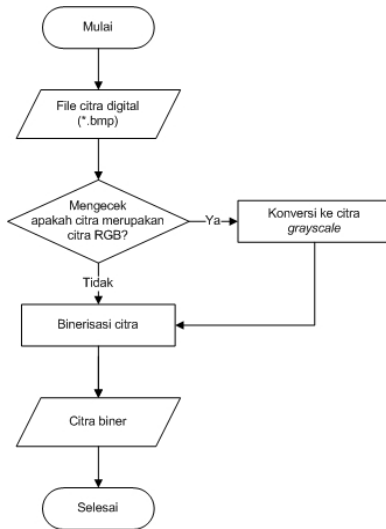
3.4.1 Pengambilan Informasi Citra Digital

Pengambilan informasi citra, yaitu proses mendapatkan informasi citra digital. Informasi citra yang diambil meliputi informasi nama citra, ukuran citra, lebar citra, tinggi citra, dan kedalaman citra.



Gambar 3.6. *Flowchart* proses pengambilan informasi citra digital.

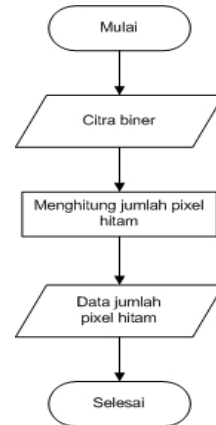
3.4.2 Preprocessing Citra Digital



Gambar 3.7. *Flowchart* proses *preprocessing* citra digital

3.4.3 Ekstraksi Fitur Citra

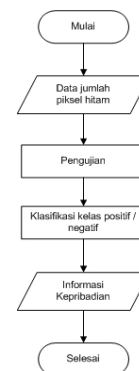
Ekstraksi fitur citra merupakan proses untuk menghitung jumlah *pixel* hitam yang ada pada citra. Data hasil ekstraksi fitur ini akan disimpan pada proses pelatihan. Sedangkan pada proses pangujian, data ini akan diproses untuk menghasilkan informasi kepribadian.



Gambar 3.8. *Flowchart* proses ekstraksi fitur citra

3.4.4 Pengujian

Proses pengujian merupakan tahap akhir dari aplikasi pengenalan tulisan tangan ini. Pada proses pengujian citra masukan akan diproses melalui perhitungan SVM, sama halnya dengan proses pelatihan, dalam proses ini digunakan fungsi kernel linear.

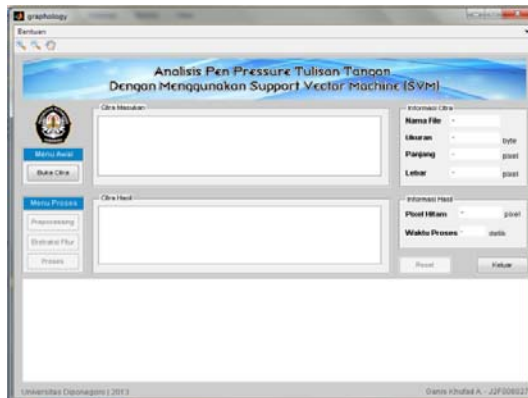


Gambar 3.9. *Flowchart* proses pengujian

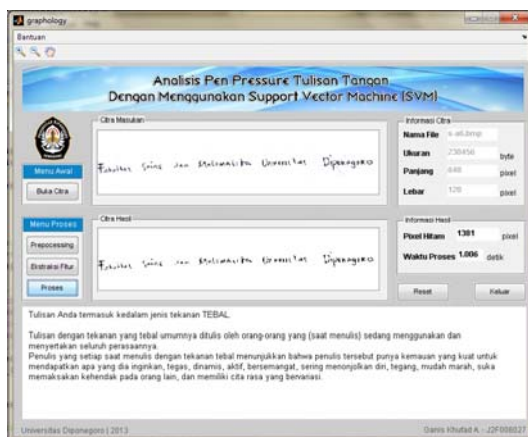
4 Implementasi dan Pengujian

4.1 Implementasi Antarmuka

Pada gambar 4.1. atau *form* utama ini terdapat informasi meliputi tombol bantuan, tombol *zoom in*, tombol *zoom out*, tombol *hand*, logo universitas, judul tugas akhir, tombol buka citra, tombol *preprocessing*, tombol ekstraksi fitur, tombol proses, sumbu citra masukan, sumbu Informasi citra, Sumbu Informasi hasil, nama universitas, tahun pembuatan aplikasi, nama pembuat, dan nomor induk mahasiswa.



Gambar 4.1. *Form* Utama



Gambar 4.2. *Form* utama setelah dilakukan proses Analisis Tulisan Tangan

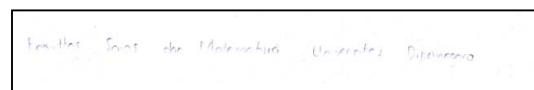
Pada gambar 4.2. dapat dilihat bahwa terdapat informasi waktu pemrosesan aplikasi Analisis Tulisan Tangan yang ditampilkan di atas panel informasi hasil. Hasil analisis tulisan

tangan yang berisi kepribadian user ditampilkan pada bagian panel hasil pengujian.

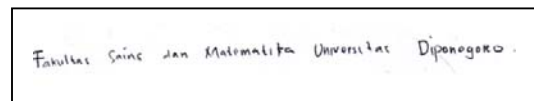
4.2 Pelaksanaan Pengujian

Selama proses pelaksanaan pengujian perangkat lunak ini, terdapat serangkaian langkah-langkah yang harus ditempuh. Proses pengujian dilakukan dengan dua jenis skenario pengujian. Skenario pertama menggunakan jumlah antara data training dan testing sama dan skenario kedua menggunakan jumlah antara data training dan testing berbeda. Sampel uji yang digunakan antara data traing dan testing berbeda. Adapun yang harus dilakukan sebelum proses pengujian adalah mempersiapkan citra uji dengan ketentuan sebagai berikut:

1. Citra uji memiliki ukuran 640 x 120 *pixel* Sebanyak 70 citra
2. Citra uji terdiri dari dua buah jenis citra yang berbeda, yaitu citra tulisan tipis (kelas positif) sebanyak 35 citra dan tebal (kelas negatif) sebanyak 35 citra. Untuk contoh citrakelas positif dan negatif lebih jelasnya dapat dilihat pada gambar 4. 3 dan 4. 4.



Gambar 4.3 Citra tulisan tangan tipis (kelas positif)



Gambar 4.4 Citra tulisan tangan tebal (kelas negatif)

3. Jenis citra yang digunakan adalah citra bitmap (*.bmp).

4.3 Hasil Pengujian

Eksperimen dilakukan sebanyak 5 kali untuk data training dan data testing yang sama, yaitu masing-masing 35 data dan sebanyak 4 kali untuk jumlah data training dan data testing

yang berbeda. Setiap eksperimen mempunyai data citra yang berbeda.

Tabel 4.1 Hasil eksperimen dengan jumlah data training dan data testing sama

Eksperimen	Data Training			Data Testing			Jumlah Data Error	Tingkat Keakuratan (%)
	K+	K-	Σ	K+	K-	Σ		
1	17	18	35	18	17	35	0	100%
2	20	15	35	15	20	35	3	91.4%
3	16	19	35	19	16	35	0	100%
4	14	21	35	21	14	35	2	94.2%
5	18	17	35	17	18	35	1	97.1%
Rata - rata								96.54%

Tabel 4.2 merupakan hasil eksperimen dengan jumlah data training tidak sama dengan data testing. Komposisi data yang digunakan dapat dilihat pada bagian lampiran.

Tabel 4.2 Hasil eksperimen dengan jumlah data training dan data testing tidak sama

Eksperimen	Data Training			Data Testing			Jumlah Data Error	Tingkat Keakuratan (%)
	K+	K-	Σ	K+	K-	Σ		
1	25	25	50	10	10	20	0	100%
2	23	22	45	12	13	25	2	95.5%
3	13	12	25	22	23	45	6	86.6%
4	10	10	20	25	25	50	7	86%

4.4 Analisis Hasil

Dari tabel 4.1 dapat diketahui bahwa rata-rata persentase keberhasilan pengenalan pola tekanan tulisan tangan sebesar 96,54%. Adanya error dalam pengklasifikasian ini kemungkinan disebabkan oleh tingkat margin data kelas positif dan negatif yang besar pada proses pelatihan, sehingga klasifikasi hasil perhitungan SVM bergeser sedikit melebihi batas ambang atau sedikit kurang dari batas ambang. Akan tetapi hal ini dapat diatasi dengan menambahkan jumlah data sampel.

Seperti terlihat pada tabel 4.2 pada eksperimen 1 dengan data training 50 sampel dan data testing 20 sampel menghasilkan tingkat akurasi 100%, eksperimen 2 dengan jumlah data training 45 dan testing 25 menghasilkan tingkat akurasi 95.5%, eksperimen 3 dengan jumlah data training 25 dan testing 45 menghasilkan tingkat akurasi 86.6%, eksperimen 4 dengan jumlah data training 20 dan testing 50 menghasilkan tingkat akurasi 86%. Hal ini mengindikasikan bahwa semakin besar data training maka semakin besar pula tingkat keakuratannya. Sebab semakin banyak data training, SVM semakin dapat membentuk *hyperlane* dengan sempurna.

5 Kesimpulan dan Saran

5.1. Kesimpulan

Kesimpulan yang dapat diambil dalam pembuatan tugas akhir ini adalah:

- Dihasilkan sebuah aplikasi Analisis Tulisan Tangan menggunakan algoritma *Support Vector Machine (SVM)*.
- Algoritma *Support Vector Machine (SVM)* cukup efektif dalam menganalisis pola tekanan tulisan tulisan untuk mengetahui kepribadian seseorang. Rata-rata tingkat keberhasilan pengenalan yang dihasilkan sebesar 96,54%.
- Jumlah *pixel* hitam pada analisis pen *pressure* tulisan tangan dapat digunakan sebagai ciri atau atribut untuk proses pengklasifikasian.

5.2. Saran

Penelitian ini masih dapat dikembangkan lebih lanjut. Beberapa saran untuk mengembangkan penelitian ini adalah sebagai berikut:

- Menambahkan jumlah sampel data untuk proses pelatihan. Karena semakin banyak data training, SVM semakin dapat membentuk *hyperlane* dengan sempurna.
- Analisis *pen pressure* tulisan tangan dapat dikembangkan dengan menggabungkan fitur lain seperti analisis ukuran huruf, jarak spasi antar kata, kemiringan tulisan,

- dan pola garis dasar tulisan, sehingga aplikasi dapat semakin *detail* dan lengkap dalam menggambarkan kepribadian seseorang.
- c. Menambahkan proses penapisan derau (*noise filtering*) agar *noise* dari cetakan kertas karbon yang bukan merupakan bagian dari tulisan dapat diminimalisir.
 - d. Proses pengujian lebih baik melibatkan seorang pakar analisis tulisan tangan (grafolog), agar hasil klasifikasi dari aplikasi dapat dibandingkan dengan hasil analisis dari grafolog.

REFERENSI

- [1] Endah, Sukmawati Nur, M. Rahmat Widyanto, 2012, "*Klasifikasi Pose Skeleton Manusia Dengan Support Vector Machine*", dalam Jurnal Ilmu Komputer dan Informasi (1979 – 0732) Volume 1-No 2.
- [2] Febrianto, Krisna, 2011, "*Penerapan Fuzzy Feature Extraction Dan Jaringan Syaraf Tiruan Resilient Propagation Dalam Pengenalan Tulisan Tangan Katakana*", diakses dari repository.upi.edu/operator/upload/s_kom_0608483_chapter1.pdf, pada tanggal 15 Juni 2012.
- [3] Imran, 2012, "*Membaca Karakter Seseorang Melalui Tulisan (Grafologi)*", diakses dari <http://sulsel.kemenag.go.id/index.php?artikel&id=13728>, pada tanggal 12 Juli 2012.
- [4] Intisari, 2001, "*Sejarah Grafologi*", diakses dari <http://www.wirasabha.web.id/sejarah-grafologi>, pada tanggal 10 Juni 2012
- [5] Ladjamudin, Al Bahra Bin, 2006, "*Rekayasa Perangkat Lunak*", Yogyakarta: Graha Ilmu.
- [6] Ludvianto, Bayu, 2012, "*Grapho for Success: Analisis Tulisan Tangan*", Jakarta: Penerbit PT. Gramedia Pustaka Utama.
- [7] Munir, Rinaldi, 2005, "*Pengolahan Citra Digital dengan Pendekatan Algoritmik*", Jakarta: Penerbit Informatika.
- [8] Moler, 2004, "*MATLAB About*", diakses dari <http://www.mathworks.com>, pada tanggal 23 Januari 2013, pukul 15.50 WIB.
- [9] Nugroho, Anto Satriyo, dkk, 2003, "*Support Vector Machine: Teori dan Aplikasi dalam Bioinformatika*", dalam Jurnal Kuliah Umum IlmuKomputer.Com, Jakarta.
- [10] Parno, _____, "*Data Flow Diagram*", diakses dari http://parno.staff.gunadarma.ac.id/Downloads/files/4403/PSI04_DFD.pdf, pada tanggal 23 Februari 2013, Pukul 20.30 WIB.
- [11] Santi, Candra Noor, 2011, "*Mengubah Citra Berwarna Menjadi Grayscale dan Citra biner*" Jurnal Teknologi Informasi DINAMIK Volume 16, No.1, Januari 2011 : 1419
- [12] Santosa, Budi, 2007, "*Data Mining: Teknik Pemanfaatan Data untuk Keperluan Bisnis*", Yogyakarta: Penerbit Graha Ilmu.
- [13] Sommerville, I, 2001, "*Software Engginering*", Jakarta: Penerbit Erlangga.
- [14] Sudarsono, _____, "*Flowchart*", diakses dari sudarsono.staff.gunadarma.ac.id/Downloads/files/16512/Flowchart.pdf, pada tanggal 27 Februari 2013, pukul 22.57 WIB.
- [15] Sugiharto, Aris, 2006, "*Pemrograman GUI dengan MATLAB*", Yogyakarta: Penerbit Andi.
- [16] Suksmono, _____, "*5. Pendahuluan Pengenalan Pola*", diakses dari <http://radar.ee.itb.ac.id/~suksmono/Lectures/ec4041/ppt/PP1.%20Pendahuluan%20Pengenalan%20Pola.pdf>, pada tanggal 16 November 2012, pukul 19.20 WIB.
- [17] Sutoyo, T,dkk. 2009, "*Teori Pengolahan Citra Digital*", Yogyakarta: Penerbit Andi

