

## **BAB 2**

### **TINJAUAN PUSTAKA**

#### **2.1.Grafologi**

Grafologi berasal dari kata *graphos* yang berarti coretan dan *logos* adalah ilmu. Berdasarkan pengertian tersebut dapat dikatakan bahwa grafologi merupakan sebuah cabang ilmu yang mempelajari sebuah coretan tangan. Tulisan tangan disini bukan selalu huruf saja yang dapat dianalisis, namun sebuah coretan-coretan tangan juga dapat memiliki arti tersendiri. Grafologi juga dikatakan sebagai seni membaca tulisan tangan, karena tulisan setiap orang memiliki ciri-ciri yang khusus [5].

Grafologi adalah seni dan ilmu yang mempelajari tentang tulisan tangan. Karena tulisan tangan berasal dari otak manusia, maka apa yang dituangkan dalam tulisan itu adalah buah pikirannya. Buah pikiran manusia ini dapat memberi gambaran atau mencerminkan kepribadian manusia. Meski kegiatan menulis ini tampak sebagai kegiatan yang dikendalikan oleh pikiran sadar, tetapi ternyata justru sangat dipengaruhi oleh pikiran bawah sadar manusia. Oleh karena itu, bentuk, gaya tulisan, dan karakter-karakter dalam diri masing-masing orang tidaklah sama.[6].

##### **2.1.1. Sejarah Grafologi**

Ilmu grafologi telah dikenal sejak berabad-abad yang lalu. Misalnya di Tiongkok, pengetahuan tentang tulisan tangansudah digunakan sejak tahun 1000 Masehi, meskipun tidak secara ilmiah. Baru pada tahun 1622, seorang dokter dari Italia, Dr. Camillo Baldi, mengemukakan penemuan tentang ilmu pengenalan tulisan yang dibuat secara sistematis/ilmiah dan dimuat dalam bukunya. Didalam buku tersebut dijelaskan tentang adanya keterikatan antara tulisan tangan seseorang dengan karakter dan kepribadiannya [6].

Lalu pada tahun 1741-1801, pastor swis, J.C Lavanter membuat laporan yang lebih sistematis dan cermat tentang tulisan tangan. Meskipun masih berupa

perkiraan dan menerangkan tentang sifat-sifat umum, tetapi telah memberikan inspirasi bagi orang lain, khususnya di Paris. Orang yang terinspirasi adalah Louis Hocquart. Selain itu, ada juga yang berasal dari kelompok agama, seperti Kardinal Regnier, Uskup Agung Cambrai, Uskup Boudinet, dan Flandrin [6].

Grafologi kemudian mengalami perkembangan pesat di Prancis dengan memunculkan tokoh-tokoh terdepan dalam bidang grafologi, diantaranya Jules Depoin, Binet G. Trade, dan Assene Aruss yang tergabung dalam "Himpunan Studi Grafologi". Namun, pusat perkembangan kemudian berpindah ke Jerman pada pertengahan abad XVIII dengan tampilnya Adlof Hentze, Sohwieland, Gerhard Wilhem Langen Burch, dan Rudolphine Poppee [6].

### **2.1.2. Manfaat Grafologi**

Berdasarkan dari sejarahnya, ilmu grafologi digunakan untuk mengenali karakter seseorang dari tulisan tangannya. Berikut adalah beberapa contoh dari manfaat grafologi : [7]

1. Mengetahui *lifetrap*.
2. Mengetahui karakter orang lain.
3. Membantu kemampuan kita dalam berkomunikasi.
4. Identifikasi.
5. Sarana intropeksi diri.
6. Meningkatkan keahlian.
7. Mengungkap kasus kejahatan

### **2.1.3. Tanda Tangan menurut ilmu Grafologi**

Tanda tangan merupakan atribut biometrik yang penting dari individu yang dapat digunakan sebagai identitas. Penggunaan tanda tangan merupakan cara yang alami dan tradisional sebagai identitas yang sah. Hal ini membuat keberadaan tanda tangan menjadi penting, sehingga diperlukan adanya sistem yang digunakan untuk memberi pengamanan supaya tidak disalah gunakan oleh pihak yang tidak bertanggung jawab .

Tanda tangan adalah hasil proses menulis seseorang yang bersifat khusus sebagai substansi simbolik. Tanda tangan merupakan bentuk yang paling banyak digunakan untuk identifikasi seseorang [3]. Pada umumnya pengenalan tanda

tangan dilakukan secara manual oleh seseorang dengan mencocokkan secara langsung tanda tangan yang sah dengan tanda tangan yang dilakukan saat itu. Contoh-contoh tanda tangan setiap orang umumnya identik namun tidak sama, artinya tanda tangan seseorang sering berubah-ubah setiap waktu. Perubahan ini menyangkut posisi, ukuran maupun faktor tekanan tanda tangan [2]. Kelemahan sistem manual dalam identifikasi tanda tangan adalah si pemeriksa tanda tangan harus teliti dalam melakukan pencocokan untuk menghindari kesalahan. Oleh karena itu untuk mengatasi kelemahan pencocokan tanda tangan secara manual, proses pencocokan tanda tangan perlu dilakukan secara otomatis dengan sistem komputer sehingga diharapkan mempermudah dalam identifikasi tanda tangan seseorang.

#### 2.1.4. 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 2.1, yaitu pola tanda tangan beserta kepribadian yang menyertai [1].

**Tabel 2.1 Fitur Pada Tanda Tangan**

Fitur	Gambar	Ciri	Kepribadian
Awal Kurva		Lengkung mundur	Nyaman akan masalah.
		Lengkung tajam	Mampu memformasikan pikiran secara tajam.
		Lengkung lembut	Hati-hati, ramah, diplomatis.
Coretan akhir		Menaik	Terbuka, pandangan kedepan, keinginan maju, percaya diri.
		Menurun	Kurang semangat, berfikir realistis, kurang percaya diri, mudah putus asa.

Fitur	Gambar	Ciri	Kepribadian
Cangkang		Lengkung tertutup	Ketakutan berlebih, introvert, tidak memperdulikan sekitar, tidak suka bergaul dan tidak suka bekerja sama.
Coretan di tengah		Adanya coretan	Kurang percaya diri dan mudah depresi.
Garis bawah		Adanya garis bawah	Mebutuhkan dukungan dalam membuat keputusan, serta memiliki kemampuan dalam memimpin.
Margin ekstrim		Cenderung ke kanan	Ceroboh, kurang perhatian
		Cenderung ke kiri	Takut gagal, takut pada orang lain, kurang percaya diri, pesimis.
		Cenderung ke atas	Respek pada diri sendiri, mencerminkan pribadi bahagia.
		Cenderung ke bawah	Depresi, pemalu, merasa asing.
Struktur titik		Tanda titik	Pendirian stabil, memiliki rasa curiga, selalu menjaga jarak tidak mudah percaya.

Fitur	Gambar	Ciri	Kepribadian
Tanda tangan terpisah		Tanda tangan terpisah	Memiliki pengalaman yang kurang menyenangkan di masa lalu.
Garis terpisah		Coretan akhir garis terpisah	Membatasi keinginannya, tidak berani mengambil resiko, sering patah semangat dan ragu mengambil keputusan.

## 2.1 Pengolahan Citra Digital

Pengolahan citra digital adalah istilah umum untuk berbagai teknik yang keberadaannya untuk memanipulasi dan memodifikasi citra dengan berbagai cara. Setiap foto dengan bentuk citra digital dapat diolah menggunakan perangkat lunak tertentu. Foto adalah contoh gambar berdimensi dua yang dapat diolah dengan mudah [7].

### 2.1.1 Pengertian Citra

Manusia merupakan makhluk visual yang mengandalkan penglihatan untuk memahami sekelilingnya, dengan kemampuan tersebut manusia dapat merasakan dan mengetahui perbedaan sesuatu hal dengan cepat. Citra sendiri didefinisikan sebagai fungsi dari dua variabel misalnya  $a_{x,y}$  di mana  $a$  merupakan amplitudo atau dalam hal ini ialah kecerahan, sedangkan  $x,y$  ialah koordinat yang membentuk citra [4].

Citra merupakan istilah lain untuk gambar pada salah satu komponen multimedia dengan karakteristik yang tidak dimiliki oleh data teks. Secara harfiah citra ialah gambar atau sebuah terusan cahaya pada bidang dwimatra (dua dimensi) yang ditinjau dari sudut pandang matematis [8]. Cahaya yang menyinari suatu objek akan memberikan gambaran tentang objek tersebut, hal ini dikarenakan adanya pantulan cahaya saat sumber cahaya menyinari objek. Pantulan cahaya tersebut akan ditangkap oleh alat – alat optik, misalnya mata manusia, kamera, *scanner*, dan sebagainya [4].

Citra secara umum terbagi menjadi dua jenis, citra analog dan citra digital, dalam studi tentang pengolahan citra, citra yang digunakan pada penelitian ini

merupakan citra digital yang dihasilkan dari mesin atau alat elektronik yang mampu menangkap gambar [9].

Pada penelitian ini, citra berperan sebagai data masukan yang akan berpengaruh terhadap proses dan hasil. Citra dengan kualitas yang buruk akan menurunkan akurasi pada klasifikasi. Oleh karena itu, guna memperoleh kualitas citra yang lebih baik, dilakukanlah pengolahan citra.

### 2.1.2 Pengolahan Citra

Citra atau gambar merupakan salah satu media yang dapat menyampaikan informasi lebih detil dan jelas bila dibandingkan dengan teks, penggunaannya pun dapat merepresentasikan suatu objek secara lebih nyata, namun tidak semua citra memiliki kualitas yang baik [4].

Dalam pengolahan citra digital terutama citra yang digunakan sebagai data masukan suatu proses, perlu memiliki kualitas yang baik. Hal-hal yang mengganggu kualitas citra contohnya adalah adanya bercak hitam atau putih yang bukan bagian dari objek pada citra.

Meskipun demikian, untuk memperoleh citra yang baik memerlukan alat optik yang bagus dengan harga yang cukup mahal. Alternatif dari penggunaan alat optik yang mahal adalah dilakukannya pengolahan citra dengan beragam macam metode, salah satunya ialah menggunakan konsep *smoothing*.

Pengolahan gambar atau pengolahan citra yang sering disebut *image processing*. Adalah suatu proses pengolahan gambar yang mengubah gambar menjadi gambar lain yang memiliki kualitas lebih baik untuk tujuan tertentu. Pengolahan citra memiliki dua tujuan utama, yaitu [7]:

1. Memperbaiki kualitas citra, dimana citra yang dihasilkan dapat menampilkan informasi secara jelas. Hal ini berarti manusia sebagai pengolah informasi (*human perception*).
2. Mengekstraksi informasi ciri yang menonjol pada suatu citra, hasilnya adalah informasi citra dimana manusia mendapatkan informasi ciri dari citra secara numerik.

Operasi-operasi pada pengolahan citra diterapkan pada citra bertujuan untuk [7]:

1. *Image enhancement* adalah perbaikan citra yang dilakukan untuk meningkatkan kualitas penampakan citra atau menonjolkan beberapa aspek informasi yang terkandung dalam citra.
2. *Image restoration* adalah menghilangkan atau meminimumkan sebuah cacat yang ada pada sebuah citra.
3. *Image segmentation* adalah operasi citra yang perlu mengelompokkan, mencocokkan atau mengukur sebuah citra.
4. *Image analysis* adalah sebuah ekstraksi ciri-ciri tertentu yang dimiliki pada sebuah citra untuk membantu dalam pengidentifikasian objek. Proses segmentasi citra kadangkala diperlukan untuk melokalisasi objek yang diinginkan dari sekelilingnya.
5. *Image reconstruction* adalah penggabungan dari beberapa citra dan menghasilkan sebuah citra baru.
6. *Image compression* adalah pemampatan sebuah citra, yaitu pengecilan ukuran citra.
7. *Steganografi* adalah penyembunyian sebuah data rahasia yang disimpan didalam sebuah citra sehingga keberadaan data tersebut tidak diketahui oleh orang lain.

Beberapa jenis operasi dari pengolahan citra adalah sebagai berikut [1]:

1. Modifikasi Kecemerlangan (*Brightness Modifikasi*)  
Mengubah nilai keabuan atau warna dari gelap menjadi terang ataupun sebaliknya mengubah citra yang terlalu terang/pucat menjadi gelap.
2. Peningkatan Kontras (*Contrast Enhancement*)  
Dengan meningkatkan kontras dari sebuah citra maka titik yang cenderung gelap menjadi lebih gelap dan yang cenderung terang menjadi lebih terang.
3. Negasi  
Operasi untuk mendapatkan citra negatif (*negative image*)
4. Mengubah ukuran(*resize*)  
*Resize* atau penskalaan adalah sebuah operasi geometri yang digunakan untuk memperbesar atau memperkecil ukuran dari sebuah citra sesuai dengan ukuran yang dibutuhkan. Pada penskalaan apabila variabel penskalaannya bernilai

lebih besar dari 1, maka ukuran citra akan diperbesar, namun apabila variabel penskalaannya bernilai lebih kecil dari 1 maka ukuran citra akan diperkecil.

Proses penskalaan dapat dilakukan dengan rumus[10]:

$$x = \frac{pb * pp}{pa} \quad (2.1)$$

Keterangan

x = Nilai piksel baris baru

pb = Ukuran panjang matriks baru

pp = Posisi piksel baris

pa = Ukuran panjang matriks lama

$$y = \frac{lb * lp}{la} \quad (2.2)$$

Keterangan

y = Nilai piksel kolom baru

lb = Ukuran lebar matriks baru

lp = Posisi piksel kolom

la = ukuran lebar matriks lama

##### 5. Pengabuan (*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*.

$$\mathbf{Grayscale} = 0,2989 * \mathbf{R} + 0,5870 * \mathbf{G} + 0,1140 * \mathbf{B} \quad (2.3)$$

##### 6. Pengambangan (*Thresholding*)

metode thresholding adalah citra biner dengan nilai intensitas piksel sebesar 0 atau 1. Setelah citra sudah tersegmentasi atau sudah berhasil dipisahkan objeknya dengan background, maka citra biner yang diperoleh dapat dijadikan sebagai masking untuk melakukan proses cropping sehingga

diperoleh tampilan citra asli tanpa background atau dengan background yang dapat diubah-ubah [11].

7. Pencerminan (*Flipping*)

Pencerminan merupakan proses menggambar citra ke bentuk kebalikan dari citra tersebut, seperti ketika orang sedang bercermin.

8. Rotasi (*Rotating*)

Rotasi adalah proses memutar koordinat citra sesuai dengan derajat yang ditentukan

9. Pemotongan (*Cropping*)

Memotong bagian yang diinginkan dari sebuah citra

10. Penskalaan (*Scaling*)

Mengubah ukuran citra menjadi lebih besar atau lebih kecil

11. Deteksi Tepi (*Edge Detection*)

Deteksi tepi adalah suatu proses yang menghasilkan tepi-tepi dari obyek dalam sebuah citra. Algoritma *Canny* berjalan dalam 5 langkah yang terpisah yaitu :

- a. Penghalusan (*Smoothing*) : Menggabungkan gambar untuk menghilangkan noise.
- b. Mencari Gradien (*Finding gradient*) : Tepian harus ditandai pada gambar memiliki gradien yang besar.
- c. *Non-maksimum-suppression* : Hanya maxima lokal yang harus ditandai  
*Ende*
- d. *Double thresholding* : Tepian yang berpotensi ditentukan oleh *thresholding*.
- e. *Edge tracking by hysteresis* : Tepian final ditentukan dengan menekan semua sisi yang tidak terhubung dengan tepian yang sangat kuat

Tahap deteksi tepi *canny* adalah sebagai berikut:

1. Langkah pertama pada deteksi tepi *Canny* adalah mengkonvolusikan hasil matriks *Grayscale* dengan tapis Gaussian.

$$\frac{1}{115} \begin{pmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{pmatrix} \quad (2.4)$$

2. Kemudian menghitung nilai tepi menggunakan operator sobel menggunakan rumus

$$G = |G_x| + |G_y|. \quad (2.5)$$

$$M_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \quad M_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}. \quad (2.6)$$

3. Setelah menemukan tepi dengan operator sobel, maka langkah selanjutnya adalah menentukan arah tepi dengan persamaan sebagai berikut :

$$\theta = \arctan \left( \frac{|G_y|}{|G_x|} \right) \quad (2.7)$$

4. Setelah arah tepi diperoleh selanjutnya mengkonversi arah tepi kedalam salah satu kategori 0, 45, 90, 135 derajat.
5. Kemudian dilakukannya penghilangan nonmaksimum, piksel yang tidak dianggap tepi dapat diubah menjadi 0.
6. Kemudian dilakukannya pengembangan hysteresis dengan melibatkan dua ambang, T1 yaitu 125 dan T2 yaitu 255.
7. Kemudian dilakukan threshold dengan persamaan sebagai berikut:

$$T(x, y) = \begin{cases} 1 & \text{if } (x, y) \geq 255 \\ 0 & \text{if } (x, y) < 255 \end{cases} \quad (2.8)$$

Metode *edge detection* akan mendeteksi semua *edge* atau garis-garis yang membentuk objek gambar dan akan memperjelas kembali pada bagian-bagian tersebut. Tujuan pendeteksian ini adalah bagaimana agar objek di dalam gambar dapat dikenali dan disederhanakan bentuknya dari bentuk sebelumnya. Metode *Canny edge detection* merupakan pengembangan dari metode dasar *edge detection*.

Perancangan sebuah prosedur dengan menerapkan langkah-langkah metode *Canny edge detection* akan menghasilkan sebuah tampilan gambar yang berbeda dengan menampilkan efek relief didalamnya. Efek relief adalah seperti sebuah tampilan batu kasar yang diukir, yaitu garis-garis kasar yang membentuk sebuah penggambaran objek di dalamnya. Efek relief terbentuk dari bayangan terang dan gelap. Kedua bayangan ini terjadi akibat adanya sorotan sinar mengenai gambar dari arah tertentu. Kelebihan dari metode *Canny* ini adalah kemampuan untuk mengurangi *noise* sebelum melakukan perhitungan deteksi tepi sehingga tepi-tepi yang dihasilkan lebih banyak [5].

### 2.1.3 Segmentasi

Dalam pengolahan citra, terkadang kita menginginkan pengolahan hanya pada obyek tertentu. Oleh sebab itu, perlu dilakukan proses segmentasi citra yang bertujuan untuk memisahkan antara objek (*foreground*) dengan *background*. Pada umumnya keluaran hasil segmentasi citra adalah berupa citra biner di mana objek (*foreground*) yang dikehendaki berwarna putih (1), sedangkan *background* yang ingin dihilangkan berwarna hitam (0). Sama halnya pada proses perbaikan kualitas citra, proses segmentasi citra juga bersifat eksperimental, subjektif, dan bergantung pada tujuan yang hendak dicapai.

Pada proses ini citra akan dipotong menjadi dua area untuk segmentasi *vertikal* dengan ukuran piksel [25x100] supaya mendapatkan fitur kurva awal dan coretan akhir, sementara dua area untuk segmentasi *horizontal* dengan piksel [50x20] supaya mendapatkan coretan tengah dan garis bawah.

### 2.1.4 Two Dimensional Linear Discriminant Analysis

Metode 2D-LDA merupakan pengembangan dari metode LDA. Alur kerja dari metode 2D-LDA hampir sama dengan metode LDA, perbedaan utama antara metode LDA dan 2D-LDA adalah cara representasi data, jika LDA data direpresentasikan menjadi *vector* atau matriks satu dimensi maka metode 2D-LDA direpresentasikan menjadi matriks dua dimensi. Langkah pertama adalah mencari *mean* dari beberapa data, misal ada dua data citra  $x_1$  dan  $x_2$  dengan persamaan 2.1 [3] [12]

$$\mu_1 = \frac{1}{N_1} \sum_{x \in \omega_1} X, \mu_2 = \frac{1}{N_2} \sum_{x \in \omega_2} X \quad (2.9)$$

Persamaan 2.9 digunakan untuk mencari matriks *covarian* dari kedua data setelah mean didapatkan [3].

$$S_i = \sum_{x \in \omega_i} (x - \mu_i)(x - \mu_i)^T \quad (2.10)$$

Kemudian setelah mendapatkan matriks *covarian*, langkah selanjutnya mencari *within class scatter matrix* dengan persamaan 2.11 [3].

$$S_w = S_1 + S_2 \quad (2.11)$$

Persamaan 2.12 digunakan untuk mencari *between class scatter matrix* [3].

$$S_B = (\mu_1 - \mu_2)(\mu_1 - \mu_2)^T \quad (2.12)$$

Kemudian membentuk matriks proyeksi ( $W$ ) yang digunakan untuk proyeksi data baru (data testing) menggunakan persamaan 2.13 [3].

$$S_w^{-1} S_B = \lambda W \quad (2.13)$$

### 2.1.5 Support Vector Machine (SVM)

SVM pertama kali diperkenalkan oleh Vapnik tahun 1992. SVM berusaha menemukan fungsi pemisah (*hyperplane*) yang optimal sebagai pemisah dua buah kelas pada *input space*.

*Hyperplane* terbaik merupakan *hiperplane* yang memiliki *margin* maksimal yang diperoleh dari alternatif garis pemisah (*discriminant boundaries*). *Margin* adalah jarak antara *hyperplane* dengan titik terdekat dari masing-masing kelas, sedangkan titik terdekat tersebut disebut *support vector* [1].

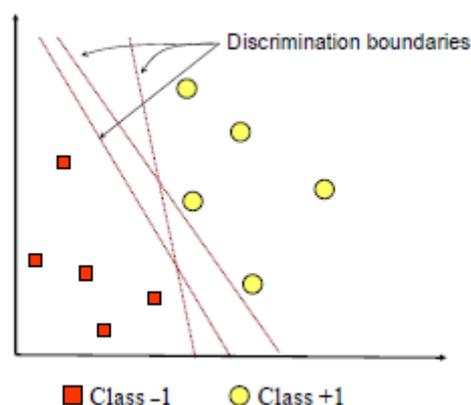
*Pattern Recognition* merupakan salah satu bidang dalam komputer *sains*, yang memetakan suatu data ke dalam konsep tertentu yang telah didefinisikan sebelumnya. Konsep tertentu ini disebut *class* atau *category* [13]. Salah satu metode yang akhir-akhir ini banyak dikembangkan sebagai *state of the art* dalam *pattern recognition* adalah SVM.

SVM dikembangkan oleh Boser, Guyon, Vapnik, dan pertama kali dipresentasikan pada tahun 1992 di *Annual Workshop on omputational Learning Theory* [13]. Konsep dasar SVM sebenarnya merupakan kombinasi harmonis dari teori-teori komputasi yang telah ada puluhan tahun sebelumnya, seperti *margin hyperplane* (Duda & Hart tahun 1973, Cover tahun 1965, Vapnik 1964, dsb.), *kernel* diperkenalkan oleh Aronszajn tahun 1950, dan demikian juga dengan

konsep-konsep pendukung yang lain. Akan tetapi hingga tahun 1992, belum pernah ada upaya merangkaikan komponen-komponen tersebut [13].

Berbeda dengan strategi *neural network* yang berusaha mencari *hyperplane* pemisah antar *class*, SVM berusaha menemukan *hyperplane* yang terbaik pada *input space*. Prinsip dasar SVM adalah *linear classifier*, dan selanjutnya dikembangkan agar dapat bekerja pada problem *non-linear*. dengan memasukkan konsep *kernel trick* pada ruang kerja berdimensi tinggi. Perkembangan ini memberikan rangsangan minat penelitian di bidang *pattern recognition* untuk investigasi potensi kemampuan SVM secara teoritis maupun dari segi aplikasi. Dewasa ini SVM telah berhasil diaplikasikan dalam problema dunia nyata (*real-world problems*), dan secara umum memberikan solusi yang lebih baik dibandingkan metode konvensional seperti misalnya *artificial neural network*. Tulisan ini memperkenalkan konsep dasar SVM, dan membahas aplikasinya di bioinformatika, yang akhir-akhir ini merupakan salah satu bidang yang berkembang cukup pesat.

Konsep SVM dapat dijelaskan secara sederhana sebagai usaha mencari *hyperplane-hyperplane* terbaik yang berfungsi sebagai pemisah dua buah *class* pada *input space*. Gambar 2.1 memperlihatkan beberapa pattern yang merupakan anggota dari dua buah *class*, yaitu *class +1* dan  $-1$  [13]:

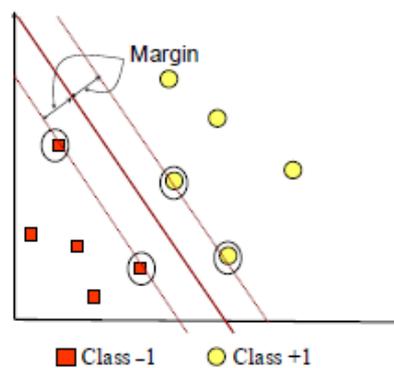


**Gambar 2.1 SVM Berusaha Menemukan *Hyperplane* Terbaik**

*Pattern* yang tergabung pada *class -1* disimbolkan dengan warna merah (kotak), sedangkan *pattern* 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*) ditunjukkan pada Gambar 2.1.

*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*, seperti terlihat pada Gambar 2.2 dimana garis yang solid menunjukkan *hyperplane* yang terbaik [13]:



**Gambar 2.2 SVM Mencari *Hyperplane* 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.

Dalam *linear Support Vector Machine* pemisah merupakan fungsi *linear*. Data latih dinyatakan oleh  $(x_i, y_i)$  dan  $x_i = \{x_1, x_2, \dots, x_i\}$  merupakan atribut (fitur) set untuk data latih kelas ke-*i*. Untuk  $y_i \in \{-1, 1\}$  menyatakan label kelas. pendefinisian persamaan suatu *hyperplane* pemisah yang dituliskan dengan [14]:

$$w \times x_i + b = 0 \quad (2.14)$$

Data  $x_i$  yang terbagi ke dalam dua kelas, yang termasuk kelas -1 (sampel negatif) didefinisikan sebagai vektor yang memenuhi pertidaksamaan (2.15) berikut.

$$w \times x_i + b < 0 \text{ untuk } y_i = -1 \quad (2.15)$$

Sedangkan yang termasuk kelas +1 (sampel positif) memenuhi pertidaksamaan (2.16) berikut.

$$w \times x_i + b > 0 \text{ untuk } y_i = +1 \quad (2.16)$$

Dimana:

$x_i$  = data *input*

$y_i$  = label yang diberikan

$w$  = nilai dari bidang normal

$b$  = posisi bidang *relative* terhadap pusat koordinat,

Parameter  $w$  dan  $b$  adalah parameter yang akan dicari nilainya. Bila label data  $y_i = -1$ , maka pembatas menjadi persamaan (2.17) berikut:

$$w \times x_i + b \leq -1 \quad (2.17)$$

Bila label data  $y_i = +1$ , maka pembatas menjadi persamaan (2.18) berikut:

$$w \times x_i + b \geq +1 \quad (2.18)$$

Margin terbesar dapat dicari dengan cara memaksimalkan jarak antar bidang pembatas kedua kelas dan titik terdekatnya, yaitu  $2/|w|$ . Hal ini dirumuskan sebagai permasalahan *quadratic programming (QP) problem* yaitu mencari titik minimal persamaan (2.19) dengan memperhatikan persamaan (2.20) berikut:

$$\min r(w) = \frac{1}{2} ||w||^2 \quad (2.19)$$

$$y_i(w \times x_i + b) - 1 \geq 0, (i = 1, \dots, n) \quad (2.20)$$

Permasalahan ini dapat dipecahkan dengan berbagai teknik komputasi. Lebih mudah diselesaikan dengan mengubah persamaan (2.19) ke dalam fungsi *Lagrangian* pada persamaan (2.21), dan menyederhanakannya menjadi persamaan (2.22) berikut:

$$L(w, b, a) = \frac{1}{2} ||w||^2 - \sum_{i=1}^n a_i (y_i (w^T x_i + b) - 1) \quad (2.21)$$

$$L(w, b, a) = \frac{1}{2} ||w||^2 - \sum_{i=1}^n a_i (y_i (w^T x_i + b) + \sum_{i=1}^n a_i) \quad (2.22)$$

Dimana  $a_i$  adalah *lagrange multiplier* yang bernilai nol atau positif ( $a_i \geq 0$ ). Nilai optimal dari persamaan (2.22) dapat dihitung dengan meminimalkan  $L$  terhadap  $w$ ,  $b$  dan  $a$ . Dapat dilihat pada persamaan (2.23) sampai (2.25) berikut:

$$\frac{\partial L}{\partial w} = w - \sum_{i=1}^n a_i y_i x_i = 0 \quad (2.23)$$

$$\frac{\partial L}{\partial w} = \sum_{i=1}^n a_i y_i = 0 \quad (2.24)$$

$$\frac{\partial L}{\partial w} = w - \sum_{i=1}^n a_i y_i (w^T x_i + b) - \sum_{i=1}^n a_i = 0 \quad (2.25)$$

Maka masalah *Lagrange* untuk klasifikasi dapat dinyatakan pada persamaan (2.26) berikut:

$$\min r(w) = \frac{1}{2} \|w\|^2 - \sum_{i=1}^n a_i y_i (w^T x_i + b) - \sum_{i=1}^n a_i \quad (2.26)$$

Dengan memperhatikan persamaan (2.27) dan (2.28) berikut:

$$w - \sum_{i=1}^n a_i y_i x_i = 0 \quad (2.27)$$

$$\sum_{i=1}^n a_i y_i = 0 \quad (2.28)$$

Model persamaan (2.26) diatas merupakan model primal *Lagrange*. Sedangkan dengan memaksimalkan L terhadap  $a_i$ , persamannya menjadi persamaan (2.29) berikut:

$$\text{Max} \sum_{i=1}^n a_i - \frac{1}{2} \sum_{i=1, j=1}^n a_i a_j y_i y_j^T x_i x_j^T \quad (2.29)$$

Dengan memperhatikan persamaan (2.30) berikut:

$$\sum_{i=1}^n a_i y_i = 0, a_i \geq 0 \quad (i, j = 1, \dots, n) \quad (2.30)$$

Untuk mencari nilai  $x_i$  dan  $y_i$  dapat dilakukan ketika sudah didapatkan nilai proyeksi baru ( $w$ ) tiap citra dari ekstrasi fitur 2D-LDA . Hasil dari ekstrasi fitur 2D-LDA diubah ke dalam bentuk format data SVM [14], sedangkan data kelas menjadi label data SVM.

Untuk mendapatkan nilai  $a_i$  , langkah pertama adalah mengubah setiap abstrak menjadi nilai vektor (*support vector*) =  $\begin{pmatrix} x \\ y \end{pmatrix}$ . Kemudian nilai vektor dari setiap citra dimasukkan ke persamaan (2.31) kernel *trick phi* berikut.

$$\varphi_1 \begin{pmatrix} x \\ y \end{pmatrix} = \begin{cases} \sqrt{Xn^2 + Yn^2} > 2, \begin{pmatrix} 4 - y + |x - y| \\ 4 - x + |x - y| \end{pmatrix} \\ \sqrt{Xn^2 + Yn^2} \leq 2, \text{ maka } \begin{bmatrix} x \\ y \end{bmatrix} \end{cases} \quad (2.31)$$

Nilai x didapatkan dari persamaan (2.32) kernel *polynomial* untuk x berikut:

$$(x \cdot y_i^T y + x)^d, x > 0 \quad (2.32)$$

Nilai y didapatkan dari persamaan (2.33) kernel *polynomial* untuk y berikut:

$$(y \cdot x_i^T x + y)^d, y > 0 \quad (2.33)$$

Untuk mendapatkan jarak tegak lurus yang optimal dengan mempertimbangkan vektor positif, maka hasil perhitungan dari substitusi nilai x dan nilai y ke persamaan (2.31) diberi nilai bias = 1 . Kemudian cari parameter  $a_i$ , dengan terlebih dahulu mencari nilai fungsi setiap citra menggunakan persamaan (2.34), lalu

mencari nilai  $a_i$  pada persamaan *Polynomial* menggunakan persamaan (2.35) dengan memperhatikan  $i,j=1,..,n$  berikut:

$$(y \cdot xi^T x + y)^d, y > 0 \quad (2.34)$$

$$(y \cdot xi^T x + y)^d = y_i \quad (2.35)$$

Setelah parameter  $a_i$  didapatkan, kemudian masukkan ke persamaan (2.36) berikut:

$$\tilde{w} = \sum_{i=1}^n a_i S_i \quad (2.36)$$

Hasil yang didapatkan menggunakan persamaan (2.36), selanjutnya digunakan persamaan (2.37) untuk mendapatkan nilai w dan b:

$$y = wx + b \quad (2.37)$$

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

Sebuah fungsi bisa menjadi fungsi *kernel* jika memenuhi *Teorema Mercer*, yang menyatakan bahwa matriks *kernel* yang dihasilkan harus bersifat semi *positive* semi *definite*. berikut ini adalah beberapa fungsi *kernel* yang umum digunakan yaitu [14]

a. *Kernel linier*

$$K(xi, x) = xi^T x$$

b. *Polynomial*

$$K(x_i, x) = (x_i, x + c)^d$$

c. *Radial Basis Function (RBF)*

$$K(xi, x) = \exp(-\gamma |xi - x|^2), \gamma > 0$$

d. *Sigmoid kernel*

$$K(xi, x) = \tanh(\gamma xi^T x + y)$$

## 2.2 Perhitungan Akurasi

Perhitungan Akurasi digunakan untuk mengetahui akurasi terhadap hasil pengujian, dalam implementasinya pengujian Akurasi menggunakan metode *Confusion Matrix*. *Confusion matrix* merupakan salah satu metode yang digunakan untuk mengukur kinerja pada saat klasifikasi. Pengujian dilakukan pada 10 kelas, diantaranya:

1. Dataset tanda tangan berjumlah 530 data

2. Data latih sebanyak 60% dan data uji 40%

Keterangan dari data yang diujikan adalah sebagai berikut: pengujian pertama dengan menggunakan data uji 40% dari data latih yang belum dilatih sebelumnya sebanyak 360 data, sementara untuk data latih menggunakan 60% dari data seluruhnya sebanyak 540 data.

Pada pengujian performansi menggunakan dua jenis proses klasifikasi diantaranya *True Positif* (TP) dan *False Negatif* (FN) untuk mendapatkan nilai akurasi yang digambarkan sebagai berikut:

**Tabel 2.2 Confussion Matrix**

Perdiksi/Kelas Sebenarnya	Kelas 1	Kelas 2	...	Kelas n
Kelas 1	TP	FN	...	FN
Kelas 2	FN	TP	...	FN
...	...	...	...	...
Kelas n	FN	FN	...	TP

Berdasarkan nilai *True Positif* dan *False Negatif* yang kemudian didapatkan nilai akurasi. Nilai akurasi menggambarkan seberapa akurat sistem dapat mengklasifikasikan data secara benar, dengan kata lain, nilai akurasi merupakan perbandingan antara data yang terklasifikasi dengan benar dengan seluruh data yang diujikan. Dan berikut adalah perhitungan akurasi yang akan digunakan dalam proses pengujian akurasi, adalah sebagai berikut [15]:

$$Akurasi = \frac{TP}{(TP+FN_1+\dots+FN_n)} \times 100\% \quad (2.38)$$

### 2.5. Flowchart

*Flowchart* (Diagram Alir) adalah bagan (*Chart*) yang menunjukkan alir (*flow*) di dalam program atau prosedur sistem secara logika. *Flowchart* merupakan metode untuk menggambarkan tahap-tahap pemecahan masalah dengan mempresentasikan simbol-simbol tertentu yang mudah dimengerti, mudah digunakan dan standar. Tujuan penggunaan *flowchart* adalah untuk menggambarkan suatu tahapan penyelesaian masalah secara sederhana, terurai, rapi, dan jelas dengan menggunakan simbol-simbol yang standar. Tahapan penyelesaian masalah yang disajikan harus jelas, sederhana, dan tepat [16].

Pada waktu akan menggambar suatu bagan alir, *programmer* dapat mengikuti pedoman-pedoman sebagai berikut:

- a. Bagan alir sebaiknya digambarkan dari atas ke bawah dan mulai dari bagian kiri suatu halaman.
- b. Kegiatan di dalam bagan alir harus ditunjukkan dengan jelas.
- c. Harus ditunjukkan dari mana kegiatan akan di mulai dan di mana akan berakhirnya.
- d. Masing-masing kegiatan di dalam bagan alir sebaiknya digunakan kata yang mewakili suatu pekerjaan.
- e. Masing-masing kegiatan di dalam bagan alir harus di dalam urutan yang semestinya.
- f. Kegiatan yang terpotong dan akan disambung di tempat lain harus ditunjukkan dengan jelas menggunakan simbol penghubung.
- g. Gunakan simbol-simbol bagan alir standar.

## **2.6. Diagram Konteks**

Diagram konteks adalah diagram yang terdiri dari suatu proses dan menggambarkan ruang lingkup suatu sistem. Diagram konteks merupakan level tertinggi dari *data flow diagram* yang menggambarkan seluruh masuk ke sistem atau keluar dari sistem. Ia akan memberi gambaran tentang keseluruhan sistem. Sistem dibatasi oleh *boundary* (dapat digambarkan dengan garis putus). Dalam diagram konteks hanya ada satu proses dan tidak boleh ada *store* dalam diagram konteks.

## **2.7. Data Flow Diagram (DFD)**

Diagram yang menggambarkan proses dari data flow diagram. *Data flow diagram* atau yang sering disebut DFD memberikan pandangan secara menyeluruh mengenai sistem yang ditangani, menunjukkan fungsi-fungsi utama atau proses yang ada, aliran data, *external entity*. DFD adalah diagram yang menguraikan proses apa yang ada dalam diagram konteks.

## **2.8. Kamus Data**

Kamus data berfungsi membantu pelaku sistem untuk mengartikan aplikasi secara detail dan mengorganisasi semua elemen data yang digunakan

dalam sistem secara persis sehingga pemakai dan penganalisa sistem mempunyai dasar pengertian yang sama tentang masukan, keluaran, penyimpanan dan proses. Kamus data sering disebut juga dengan sistem *data dictionary* adalah katalog fakta tentang data dan kebutuhan-kebutuhan informasi dari suatu sistem informasi. Atribut yang berfungsi sebagai *key* juga dibedakan dengan yang bukan *key* dengan menggarisbawahi atribut tersebut

### **2.9. Pengujian *White Box***

Pengujian dilakukan untuk menemukan kesalahan dan kekurangan yang tidak ditemukan sebelumnya, pengujian ini dilakukan khususnya pada pengujian fungsionalitas serta kinerja aplikasi yang telah dibangun. Pengujian pada penelitian ini meliputi skenario pengujian, hasil pengujian dan evaluasi pengujian.