

BAB 2

TINJAUAN PUSTAKA

2.1 Grafologi

Grafologi salah satu cabang ilmu yang dipelajari oleh bidang keilmuan psikologi, khususnya dalam mata kuliah psikografik atau psikodiagnostik. Grafologi merupakan studi dan analisis mengenai tulisan tangan, terutama yang berhubungan dengan psikis manusia dan dengan mempelajari tulisan tangan kita dapat menginterpretasi karakter seseorang. Menurut para ahli, grafologi adalah ilmu yang mempelajari tulisan tangan, yang selama ratusan tahun telah diakui sebagai hal unik yang dimiliki oleh masing-masing individu [6].

Grafologi diambil dari dua kata yaitu "*graphos*" yang mempunyai arti coretan dan "*logos*" yang mempunyai arti ilmu. Berdasarkan pada pengertian tersebut, maka dapat dikatakan bahwa grafologi merupakan sebuah cabang ilmu yang mempelajari sebuah coretan tangan. Tulisan tangan bukan berarti selalu mengenai huruf saja yang dapat dianalisis, tetapi coretan tangan juga mempunyai arti tersendiri [6].

2.1.1 Sejarah dan Definisi Grafologi

Ilmu yang mempelajari tentang tulisan tangan disebut grafologi. Grafologi merupakan ilmu tentang aksara atau sistem tulisan, ilmu surat tangan, ilmu tentang hubungan antara watak dan tulisan tangan (rajab). Tulisan tangan adalah kegiatan tertua di dunia yang sudah dikenal sejak berabad-abad yang lalu yakni pada tahun 1000 Masehi [6]. Pada tahun 1622, dokter dari Itali bernama Camillo Baldi mengemukakan penemuan ilmu tentang pengenalan penulisan yang dibuat secara sistematis yang dimuat dalam bukunya. Dalam bukunya, diterangkan adanya hubungan yang unik antara tulisan tangan seseorang dengan karakter dan kepribadian. Kata "*Grafologi*" baru pertama kali diungkapkan oleh Jean Michon yang berasal dari Prancis pada abad ke-19. Kata tersebut lahir dari bahasa Yunani "*grapho*" yang berarti menulis dan "*logos*" yang berarti ilmu. Tahun 1741-1801,

Pastor dari Swiss yang bernama J.C. Lavanter mencoba membuat laporan yang sistematis dan cermat mengenai tulisan tangan, namun masih berupa perkiraan saja. Pada laporannya dia menjelaskan tentang sifat-sifat umum yang dimiliki manusia dari tulisan tangan. Hasil interpretasi yang dilahirkan oleh tokoh-tokoh sebelumnya kemudian dikembangkan kembali oleh Abbe Jean Hyppolyte Michon yang dibantu oleh kedua asistennya yaitu Debarolle dan Dellestre, berhasil menghimpun sebuah katalog yang penuh dengan tanda garis dan ciri-ciri keteraturannya. Pada tahun 1858-1940, teori Michon dibantah oleh muridnya yang bernama Jules Crepieux-Jamin. Menurutnya banyak ciri-ciri bawaan yang tidak dapat diungkapkan secara langsung hanya melalui penggambaran grafis yang sederhana, tetapi harus dilihat juga dari beberapa rangkaian karakteristik. Pada tahun 1930, grafologi mulai dipelajari di klinik psikologi Harvard oleh salah satu tokoh psikologi yaitu Gordon Allport. Selanjutnya tahun 1955, tokoh lain yaitu Klara Roman dan George Staemphli mulai mengembangkan factor-faktor penting untuk menilai karakter seseorang dilihat dari tulisan tangan.

2.1.2 Manfaat Grafologi

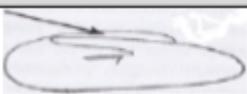
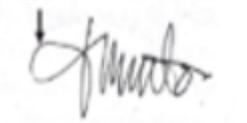
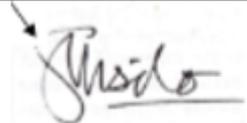
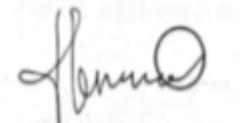
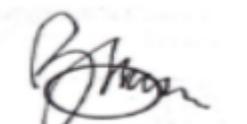
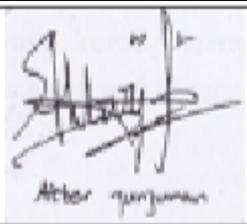
Berdasarkan dari sejarahnya, ilmu grafologi digunakan untuk mengetahui dan membaca karakter dan kepribadian seseorang dari tulisan tangan. Berikut ini adalah contoh dari manfaat [6]

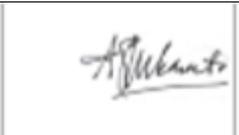
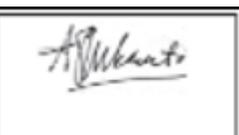
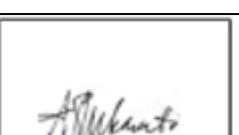
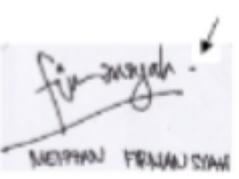
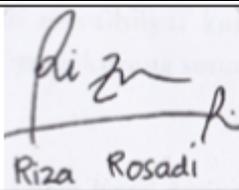
1. Mengetahui karakter orang lain.
2. Mengetahui *lifetraps*.
3. Sarana introspeksi diri.
4. Membantu kemampuan kita dalam berkomunikasi.
5. Meningkatkan keahlian.
6. Identifikasi
7. Mengungkap kasus kejahatan.

2.1.3 Dasar Penentuan Analisis Tanda Tangan

Terdapat fitur-fitur dasar yang dapat digunakan untuk menganalisis tanda tangan, berikut adalah dasar-dasar penentuan analisis tanda tangan [10].

Tabel 2.1 Tabel fitur tanda tangan [10]

No	Fitur	Gambar	Ciri	Kepribadian
1	Cangkang		Lengkung tertutup	Ketakutan berlebihan, introvert, tidak memperdulikan sekitar, tidak suka bergaul dan bekerja sama
2	Awal kurva		Lengkung mundur	Nyaman akan masa lalu
			Lengkung tajam	Mampu memformulasi pikiran secara tajam
			Lengkung lembut	Hati-hati, ramah, diplomatis
3	Coretan akhir		Menaik	Terbuka, pandangan kedepan. Keinginan maju, percaya diri
			Menurun	Kurang semangat, berfikir realistis, kurang percaya, mudah putus asa
4	Coretan di tengah		Adanya coretan	Kurang percaya diri dan mudah depresi

5	Garis bawah		Adanya garis bawah	Membutuhkan dukungan membuat keputusan, serta memiliki keandalan dalam memimpin
6	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
7	Struktur titik		Ada titik	Pendirian stabil, memiliki rasa curiga, selalu menjaga jarak, tidak mudah percaya
8	Tanda tangan terpisah		Ada tanda tangan terpisah	Memiliki pengalaman kurang menyenangkan di masa lalu

9	Garis terpisah		Ada garis terpisah	Membatasi keinginannya, tidak berani mengambil resiko, sering patah semangat dan ragu mengambil keputusan.
---	----------------	---	--------------------	--

2.2 Pengolahan Citra Digital

Pengolahan Citra Digital adalah istilah umum untuk teknik memanipulasi dan memodifikasi citra dengan cara tertentu. Setiap foto dengan bentuk citra digital akan diolah menggunakan perangkat lunak sehingga menghasilkan kualitas yang lebih baik.

2.2.1 Pengertian Citra Digital

Citra Digital merupakan sebuah larik (array) yang berisi nilai-nilai real maupun kompleks yang direpresentasikan dengan deretan bit tertentu. Suatu citra dapat didefinisikan sebagai fungsi $f(x,y)$ berukuran M baris dan N kolom, dengan x dan y adalah sebuah koordinat spasial, dan amplitude f di titik koordinat (x,y) dinamakan intensitas atau tingkat keabuan dari citra pada titik tersebut. Apabila nilai x , y dan nilai amplitude f secara keseluruhan berhingga (finite) dan bernilai diskrit maka dapat dikatakan citra tersebut citra digital [7].

Nilai suatu pixel mempunyai nilai dalam rentang tertentu, dari nilai minimum sampai nilai maksimum. Berikut adalah jenis-jenis citra berdasarkan pada nilai pixelnya. [7]

1. Citra Biner

Citra biner adalah citra digital yang hanya memiliki dua kemungkinan nilai pixel yaitu hitam dan putih. Hanya dibutuhkan 1 bit untuk mewakili nilai setiap pixel dari citra biner.

2. Citra Grayscale

Citra Grayscale merupakan citra digital yang hanya memiliki satu nilai kanal pada setiap pixelnya, dengan kata lain nilai bagian RED = GREEN = BLUE. Nilai-nilai tersebut digunakan untuk menunjukkan tingkat intensitas. Warna yang dimiliki yaitu hitam, keabuan dan putih.

2.2.2 Pengolahan Citra

Pengolahan Citra atau Image Processing memiliki input dan output berupa citra [7]. Sebagai contoh, suatu citra ditransformasi ke dalam bentuk citra yang lainnya dengan tujuan untuk memperbaiki kualitas citra agar lebih baik. Untuk mengekstraksi citra

2.2.3 Grayscale

Citra grayscale atau citra berskala keabuan. Berguna untuk merubah citra berwarna menjadi citra berskala keabuan. Secara umum perubahan citra berwarna menjadi citra keabuan dengan menggunakan rumus

$$I = a \times R + b \times G + c \times B$$

Dimana (R) adalah nilai pada warna merah, (G) adalah nilai pada warna hijau, dan (B) adalah nilai pada warna biru. Sementara untuk (a, b, c) adalah nilai mutlak, nilai mutlak yang biasa dipakai untuk (a, b, c) adalah

$$a = 0.2989$$

$$b = 0.5870$$

$$c = 0.1141$$

Sehingga didapatkan rumus untuk merubah citra berwarna menjadi citra berskala keabuan adalah sebagai berikut [11]

$$I = 0.2989 \times R + 0.5870 \times G + 0.1141 \times B \quad (2.1)$$

2.2.4 Deteksi tepi (edge detection)

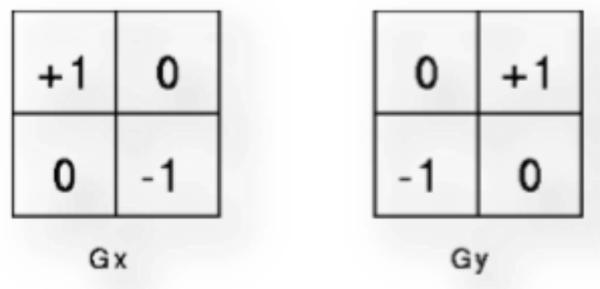
Deteksi tepi pada citra merupakan proses yang menghasilkan tepi dari objek citra, tepi-tepi ini menandakan bagian detail dari citra tersebut. Suatu titik (x, y) dikatakan sebagai tepi dari suatu citra bila titik tersebut mempunyai perbedaan yang

tinggi dengan titik tetangganya. Jenis-jenis metode dalam proses deteksi tepi adalah sebagai berikut [11].

1. Metode Robert

Metode ini mendeteksi tepi dengan cara mencari perbedaan pada arah horizontal dan arah vertical, dengan ditambahkan proses konversi biner setelah dilakukan perbedaan. agar mendapatkan tepi yang baik, konversi biner dilakukan dengan meratakan distribusi warna hitam dan putih. Kelebihan dari metode ini adalah operator yang digunakan mudah sehingga mempercepat dan memudahkan proses perhitungan pada pengukuran gradien dua dimensi spasial pada gambar.

Pada gambar 2.1 digambarkan mengenai operator Robert terdiri dari kernel 2 x 2 konvolusi. Satu kernel yang lainnya hanya diputar 90°.



Gambar 2.1 Metode Robert

Kernel ini dirancang untuk merespon secara maksimal untuk tepi berjalan pada 45° ke jaringan pixel, satu kernel untuk masing-masing dua orientasi tegas lurus. Kernel dapat diterapkan secara terpisah dengan citra masukan untuk menghasilkan pengukuran yang terpisah dari komponen gradien disetiap orientasi (G_x dan G_y). Besarnya gradien diberikan oleh

$$|G| = \sqrt{G_x^2 + G_y^2} \quad (2.2)$$

Keterangan :

G = Besaran Gradien

G_x= Gradien x

G_y= Gradien y

Walaupun biasanya, sebuah besaran perkiraan dihitung dengan menggunakan persamaan dibawah yang lebih cepat untuk menghitung

$$|G| = |G_x| + |G_y| \quad (2.3)$$

Keterangan :

G = Besaran Gradien

G_x = Gradien x

G_y = Gradien y

Sudut orientasi tepi menimbulkan gradien spasial (relatif terhadap orientasi grid pixel) diberikan oleh

$$\theta = \arctan \left(\frac{G_x}{G_y} \right) - \frac{3\pi}{4} \quad (2.4)$$

Keterangan :

θ = sudut (arah tepi)

G_x = Gradien x

G_y = Gradien y

2. Metode Prewitt

Metode ini adalah pengembangan metode robert dengan menggunakan filter HPF yang diberi satu angka nol penyangga. Metode ini mengambil prinsip dari fungsi laplacian yang dikenal sebagai fungsi untuk membangkitkan HPF. Sehingga tepi-tepi yang dihasilkan lebih banyak dari metode robert. Pada gambar 2.2 digambarkan matriks konvolusi dari metode prewitt.

-1	0	+1
-1	0	+1
-1	0	+1

G_x

+1	+1	+1
0	0	0
-1	-1	-1

G_y

Gambar 2.2 Metode Prewitt

Filter HPF menciptakan sebuah gambar dimana tepi (perubahan tajam nilai tingkat keabuan) yang akan ditampilkan. Hanya filter dengan ukuran 3 x 3 yang dapat digunakan dengan filter ini. Filter ini menggunakan dua template 3 x 3 untuk menghitung nilai prewitt gradien seperti yang ditunjukkan dibawah ini

Template

$$\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \text{ sebagai } x \quad \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} \text{ sebagai } y \quad (2.5)$$

Dimana a1 ... a9 adalah tingkat keabuan dari setiap pixel di jendela filter.

$$x = -1*a1 + 1*a3 - 1*a4 + 1*a6 - 1*a7 + 1*a9$$

$$y = 1*a1 + 1*a2 + 1*a3 - 1*a7 - 1*a8 + 1*a9 \quad (2.6)$$

Prewitt Gradient

$$G = \sqrt{G_x^2 + G_y^2} \quad (2.7)$$

Keterangan :

G = Besaran Gradien

G_x = Gradien x

G_y = Gradien y

3. Metode Sobel

Metode ini juga merupakan pengembangan metode robert. Kelebihan dari metode sobel ini adalah kemampuan untuk mengurangi noise sebelum melakukan perhitungan deteksi tepi sehingga tepi-tepi yang dihasilkan lebih banyak dibanding dua metode sebelumnya.

-1	0	+1
-2	0	+2
-1	0	+1

G_x

+1	+2	+1
0	0	0
-1	-2	-1

G_y

Gambar 2.3 Metode Sobel

Pada gambar 2.3 digambarkan bahwa operator sobel terdiri dari kernel 3 x 3 konvolusi. Kernel ini dirancang untuk merespon secara maksimal untuk tepi berjalan secara vertikal dan horizontal relatif terhadap grid pixel, satu kernel untuk masing-masing dua orientasi tegak lurus. Kernel dapat diterapkan secara terpisah dengan citra masukan, untuk menghasilkan pengukuran terpisah dari komponen gradien disetiap orientasi (G_x dan G_y). Kemudian dapat digabungkan bersama-sama untuk mencari besarnya nilai mutlak gradien pada setiap titik dan orientasi gradien itu. Besarnya gradien diberikan oleh

$$|G| = \sqrt{G_x^2 + G_y^2} \quad (2.8)$$

Keterangan :

G = Besaran Gradien

G_x = Gradien x

G_y = Gradien y

Walaupun biasanya, sebuah besaran perkiraan dihitung dengan menggunakan persamaan dibawah yang lebih cepat untuk menghitung

$$|G| = |G_x| + |G_y| \quad (2.9)$$

Keterangan :

G = Besaran Gradien

G_x = Gradien x

G_y = Gradien y

Sudut orientasi tepi menimbulkan gradien spasial (relatif terhadap orientasi grid pixel) diberikan oleh:

$$\theta = \arctan\left(\frac{G_x}{G_y}\right) \quad (2.10)$$

Keterangan :

θ = Sudut (arah tepi)

$G_x =$ Gradien x

$G_y =$ Gradien y

4. Metode Canny

Metode canny merupakan pengembangan dari metode sobel. Perbedaan metode sobel dengan metode canny terletak pada penambahan algoritma pada metode canny. Pada metode canny mempertahankan semua maxima lokal dalam gambar gradien dengan cara:[11]

- a. Putar arah gradien θ ke arah 45' terdekat, kemudian hubungkan dengan 8 titik tetangga yang terhubung dengannya.
- b. Bandingkan nilai piksel tepian saat ini dengan piksel tepian dalam arah positif dan negatif gradien. Jika arah gradien adalah utara ($\theta = 90^\circ$), bandingkan dengan piksel ke utara dan selatan.
- c. Jika nilai piksel tepian saat ini adalah yang terbesar, maka simpan nilai tepian tersebut, namun jika bukan, hapus nilai tersebut.

Operator Canny dikemukakan oleh John F. Canny tahun 1986, dikenal sebagai deteksi tepi paling optimal. Algoritma ini memberikan tingkat kesalahan yang rendah, melokalisasi titik tepi (jarak pixel tepi yang ditemukan deteksi dan tepi yang sesungguhnya sangat pendek), dan hanya memberikan satu tanggapan untuk satu tepi.[16]

Terdapat langkah-langkah yang digunakan untuk melakukan deteksi tepi canny, yaitu sebagai berikut.

1. *Image Smoothing*

Suatu proses untuk mengaburkan gambar untuk menghilangkan derau. Proses ini dapat dilakukan dengan menggunakan filter gaussian yang dapat dilihat sebagai berikut.

$$\frac{1}{159} \begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix} \quad (2.11)$$

2. Finding Gradient

Suatu proses untuk mendapatkan kekuatan tepi. Tepian harus ditandai pada gambar memiliki gradien yang besar. Operator yang digunakan untuk menentukan gradien dapat menerapkan operator sobel sebagai berikut.

$$M_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad M_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad (2.12)$$

Hasil dari gaussian filter tahap pertama deteksi tepi canny dikonvolusi kembali dengan filter operator sobel M_x sehingga menghasilkan nilai G_x dan untuk mengetahui nilai G_y didapat dengan cara melakukan konvolusi hasil gaussian dengan filter sobel M_y . Magnitudo gradien (kekuatan tepi) dapat ditentukan sebagai jarak Euclidean yang diukur dengan menerapkan hukum Pythagoras seperti berikut ini

$$G = \sqrt{G_x^2 + G_y^2} \quad (2.13)$$

Keterangan :

G = Besaran Gradien

G_x = Gradien x

G_y = Gradien y

Selanjutnya menentukan tepian yang sebenarnya, arah tepian harus ditentukan dan disimpan seoerti berikut ini.

$$\theta = \arctan\left(\frac{|G_x|}{|G_y|}\right) \quad (2.14)$$

Keterangan :

θ = sudut (arah tepi)

G_x = Gradien x

G_y = Gradien y

Aturan konversi yang berlaku sebagai berikut

- a. Semua arah tepi yang berkisar antara 0 dan 22,5 serta 157,5 dan 180 derajat (warna biru) diubah menjadi 0 derajat.

- b. Semua arah tepi yang berkisar antara 22,5 dan 67,5 derajat (warna kuning) diubah menjadi 45 derajat.
- c. Semua arah tepi yang berkisar antara 67,5 dan 112,5 derajat (warna merah) diubah menjadi 90 derajat.
- d. Semua arah tepi yang berkisar antara 112,5 dan 157,5 derajat (warna hijau) diubah menjadi 135 derajat [16]

3. *Non-maximum Suppression*

Proses yang digunakan untuk melakukan penghilangan *non-maximum*. Penghilangan ini dilakukan di sepanjang tepi pada arah tepi dan menghilangkan pixel-pixel (pixel diatur menjadi 0) yang tidak dianggap tepi. [16]

4. *Double Thresholding*

Proses tepian yang berpotensi ditentukan oleh thresholding. Pada proses ini menggunakan dua ambang T1 (ambang bawah) dan T2 (ambang atas). Semua piksel yang bernilai lebih besar dari T1 dianggap sebagai pixel tepi, lalu semua pixel yang memiliki nilai lebih besar dari T2 juga dianggap sebagai pixel tepi. [16]

5. *Edge Tracking by Hysteresis*

Merupakan proses tepian final ditentukan dengan menekan semua sisi yang tidak terhubung dengan tepian yang sangat kuat. Pengembangan hysteresis dilakukan dengan melibatkan dua ambang T1 dan T2. Nilai yang kurang dari T1 akan diubah menjadi warna hitam (nilai 0) dan nilai yang lebih besar dari T2 akan diubah menjadi putih (nilai 255). Sementara nilai yang lebih atau sama dengan T1 tetapi kurang dari T2 akan diberi nilai 128 dan yang menyatakan nilai abu-abu atau belum jelas, akan dijadikan 0 atau 255, apabila kondisi seperti itu terpenuhi, angka 128 diubah menjadi 255. [16]

$$Edge\ Tracking = \begin{cases} 0 & , nilai < T1 \\ 255 & , nilai > T2 \\ 255 & , nilai \geq T1\ dan\ nilai \leq T2 \end{cases}$$

2.3 Segmentasi Objek

Dalam pengolahan citra, terkadang pengolahan yang diinginkan hanya pada suatu objek tertentu. Oleh sebab itu, perlu dilakukan proses segmentasi citra dengan bertujuan untuk memisahkan antara objek (*foreground*) dengan *background*. Secara umum keluaran hasil segmentasi citra adalah berupa citra biner di mana objek (*foreground*) berwarna putih (1), sedangkan *background* berwarna hitam (0). Sama seperti pada proses perbaikan kualitas citra, proses segmentasi citra juga bersifat eksperimental, subjektif, dan bergantung pada tujuan yang hendak dicapai. [11]

2.4 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 [17]:

$$x = \frac{pb * pp}{pa} \quad (2.15)$$

Keterangan:

x = Nilai piksel baris baru

pb = Ukuran panjang matriks baru

pp = Posisi piksel baris

pa = Ukuran panjang matriks lama

$$y = \frac{lb * lp}{la} \quad (2.16)$$

Keterangan:

y = Nilai piksel kolom baru

lb = Ukuran lebar matriks baru

lp = Posisi piksel kolom

la = ukuran lebar matriks lama

2.5 Ekstraksi Ciri

Ekstraksi ciri merupakan proses pengukuran terhadap data yang telah di normalisasi untuk membentuk nilai sebuah fitur. Nilai fitur digunakan untuk pengklasifikasi supaya dapat mengenali unit masukan dengan unit target keluaran dan memudahkan pengklasifikasian.

2.5.1 *Principal Component Analysis*

Principal Component Analysis (PCA) adalah sebuah cara untuk mengidentifikasi pola pada data dan kemudian mengekspresikan data tersebut ke bentuk yang lain untuk menunjukkan perbedaan dan persamaan antar pola. Tujuan dari PCA adalah untuk mereduksi dimensi yang besar dari ruang data (observed variables) menjadi dimensi yang lebih kecil dari ruang fitur (independent variables), yang dibutuhkan untuk mendeskripsikan data lebih sederhana. Principal Component Analysis menggunakan vektor-vektor yang disebut dengan eigenvector dan nilai-nilai yang disebut dengan eigenvalue untuk mendapatkan fitur yang paling signifikan pada dataset. Prinsip dasar dari algoritma Principal Component Analysis adalah mengurangi satu set data namun tetap mempertahankan sebanyak mungkin variasi dalam set data tersebut.

Secara matematis Principal Component Analysis mentransformasikan sebuah variabel yang berkorelasi ke dalam bentuk yang bebas tidak berkorelasi. [20]. Berikut adalah tahap dalam algoritma PCA.

1. Hitung rata-rata seluruh sampel data diperoleh dengan menggunakan persamaan

$$\bar{x}_j = \frac{\sum_{ij}^n x_{ij}}{n} \quad (2.17)$$

Keterangan :

\bar{x} = rata-rata

n = banyak data

x = matriks data

2. Adjusted data (data yang telah disesuaikan) adalah hasil pengurangan dari setiap data dengan rata-rata setiap data yang diperoleh dengan rumusan berikut ini

$$x' = x_{ij} - \bar{x}_j \quad (2.18)$$

Keterangan :

x' = Adjusted Data

\bar{x} = rata-rata

x = matriks data

3. Hitung matrik kovarian (c) dihitung dengan menggunakan persamaan berikut

$$c = \frac{1}{M-1} x'^T \times x' \quad (2.19)$$

Keterangan:

x' = Adjusted Data

x'^T = transpose dari matrik x

M = banyak baris

4. Hitung nilai eigen dan vector eigen dari matrik kovarian dihitung dengan menggunakan persamaan karakteristik berikut ini

$$cv = \lambda Iv$$

$$cv - \lambda Iv = 0 \quad (2.20)$$

$$(c - \lambda I)v = 0$$

$$|c - \lambda I| = 0 \quad (2.21)$$

Keterangan:

c = matrik kovarian,

I = matrik Identitas

λ = nilai eigen

v = vector eigen.

5. Hitung nilai eigen yang terbesar yang berkorespondensi terhadap nilai vector eigen yang terbesar dipilih menjadi Principal Component. Vektoreigen yang disusun dari yang terbesar ke yang terkecil dipilih menjadi vektor fitur.

$$v = (eig_1, eig_2, eig_3, \dots, eig_n) \quad (2.22)$$

6. Untuk mencari Principal Component dengan \bar{x} sebagai rata-rata

$$PC = X' x v \quad (2.23)$$

Keterangan:

PC = Principal Component

\bar{x} = rata-rata

2.6 Kecerdasan Buatan

Kecerdasan buatan (Artificial Intelligence) adalah bagian dari ilmu komputer yang mempelajari bagaimana membuat mesin (komputer) dapat melakukan pekerjaan seperti dan sebaik yang dilakukan oleh manusia bahkan bisa lebih baik dari yang dilakukan manusia [12]. Menurut beberapa ahli definisi dari kecerdasan buatan sebagai berikut

1. Menurut Alan Turing, 1950

“Jika komputer tidak dapat dibedakan dengan manusia saat berbicara melalui terminal komputer, maka bisa dikatakan komputer itu cerdas, mempunyai kecerdasan.”

2. Menurut John McCarthy, 1956

“Untuk mengetahui dan memodelkan proses-proses berpikir manusia dan mendesain mesin agar dapat menirukan perilaku manusia”

3. Menurut Herbert Alexander Simon, 2001

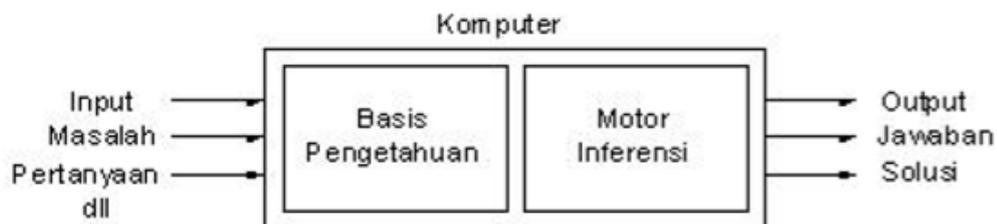
“Kecerdasan buatan merupakan kawasan penelitian, aplikasi dan instruksi yang terkait dengan pemrograman komputer untuk melakukan sesuatu hal yang dalam pandangan manusia adalah cerdas”.

4. Menurut Rich dan Knight, 1991

“Kecerdasan buatan merupakan sebuah studi tentang bagaimana membuat komputer melakukan hal-hal yang pada saat ini dapat dilakukan lebih baik oleh manusia.”

Jadi dapat disimpulkan Kecerdasan buatan (*Artificial Intelligence*) berasal dari bahasa Inggris yang terdiri dari dua kata. Yang pertama *intelligence* yang berarti cerdas, sedangkan *artificial* berarti buatan. Cerdas adalah memiliki pengetahuan dan pengalaman, penalaran yaitu bagaimana membuat keputusan dan mengambil tindakan, serta moral baik. Cerdas yang dimaksud ini merujuk kepada mesin yang bisa berpikir, menimbang tindakan serta mengambil sebuah keputusan seperti yang dilakukan oleh manusia. Pada gambar 2.4 digambarkan bagaimana membuat mesin memiliki kecerdasan, maka terdapat dua bagian utama yang dibutuhkan dalam penerapannya yaitu [12]

- 1 Basis pengetahuan (*knowledge base*): berisi fakta-fakta, teori, pemikiran dan hubungan antara satu dengan yang lainnya.
- 2 Motor inferensi (*inference engine*): kemampuan untuk menarik kesimpulan berdasarkan pengetahuan yang diberikan.



Gambar 2.4 Penerapan Kecerdasan Buatan

2.7 Machine Learning

Machine Learning merupakan sebuah teknik untuk melakukan inferensi terhadap data dengan pendekatan matematis. Inti dari *machine learning* adalah untuk membuat model (matematis) yang merefleksikan pola-pola data [13].

2.8 Supervised Learning

Supervised Learning adalah pembelajaran terarah atau terawasi. Artinya, pada pembelajaran ini proses pembelajaran, pada satu pola masukan akan diberikan ke satu neuron dan akan dikirimkan ke seluruh lapisan masukan hingga mencapai lapisan keluaran. Lapisan keluaran ini akan membangkitkan pola keluaran yang akan dicocokkan dengan pola keluaran targetnya. Jika terjadi perbedaan antara pola keluaran dengan pola yang sudah dipelajari, maka akan muncul sebuah error. Apabila nilai eror yang muncul masih sangat besar, ini mengindikasikan bahwa masih perlu dilakukan lebih banyak pembelajaran lagi [14].

2.9 Relevance Vector Machine

Pada *supervised learning*, sekumpulan data latih yang terdiri dari himpunan vektor $\{x_n\}_{n=1}^N$ digunakan bersamaan dengan target yang sesuai $\{t_n\}_{n=1}^N$ dimana pada dasarnya target merupakan nilai sebenarnya dalam hal regresi dan label kelas pada proses klasifikasi. Tujuan dari *supervised learning* adalah membuat sebuah model dari target dengan masukan dari *training set*, sehingga prediksi yang akurat terhadap target t dapat dicapai untuk nilai x yang belum diketahui sebelumnya [21]. Secara umum prediksi tersebut dapat dihitung dengan menggunakan persamaan $y(x)$, yaitu

$$y(x; w) = \sum_{i=1}^M w_i \phi_i(x) + w_0 = w^T \varphi(x) + w_0 \quad (2.24)$$

Keterangan:

w =vektor bobot (*weight vector*)

$\phi_i(x)$ = fungsi kernel terhadap data x ,

w_0 = *bias*.

Proses belajar atau pelatihan pada dasarnya merupakan sebuah proses untuk menentukan parameter dari fungsi $y(x)$, dalam konteks ini menentukan parameter bobot $w = (w_1, w_2, w_3, \dots, w_m)$ Untuk himpunan dari pasangan data latih $\{x_n, t_n\}_{n=1}^N$ tugas dari proses pelatihan ini adalah untuk mencari nilai dari bobot

$w = (w_1, w_2, w_3, \dots, w_m)^T$ sehingga fungsi $y(x)$ dapat digeneralisasi secara cukup baik terhadap data yang baru, dan bobot yang dihasilkan memiliki elemen nilai bukan nol (*nonzero*) yang sedikit. Memiliki beberapa elemen nilai yang bukan nol dapat menghasilkan representasi vektor yang *sparse*, dengan keuntungan dapat menyediakan implementasi yang cepat [21].

Relevance Vector Machine adalah sebuah metode pembelajaran mesin yang diperkenalkan oleh Mike Tipping pada tahun 2001, yang diadaptasi dari *Bayesian Framework* dan memiliki bentuk model fungsi yang mirip dengan *Support Vector Machine* [15]. Sama halnya dengan *Support Vector Machine*, *Relevance Vector Machine* digunakan dalam proses klasifikasi dan regresi. *Relevance Vector Machine* diperkenalkan untuk menutupi beberapa kelemahan yang dimiliki oleh *Support Vector Machine*. Kelemahan tersebut antara lain [15].

- a. Walaupun *Support Vector Machine* menghasilkan vektor yang relatif *sparse*, namun jumlah *Support Vector* yang ada akan secara linear bertambah seiring dengan ukuran himpunan data latih.
- b. Prediksi yang tidak bersifat probabilistik, dan *Support Vector Machine* tidak cocok untuk proses klasifikasi dimana peluang *posterior* dari keanggotaan kelas dianggap penting.
- c. *Support Vector Machine* memerlukan proses estimasi parameter *error/margin* C , yang secara umum memerlukan prosedur *cross-validation* yang dapat memperlambat proses komputasi.
- d. Fungsi *kernel* yang digunakan dalam *Support Vector Machine* harus memenuhi kondisi *Mercer*.

Pada klasifikasi untuk dua label kelas (*binary classification*), semua target dapat diklasifikasikan ke dalam dua kelas, yang dapat direpresntasikan dengan nilai 0 dan 1, seperti $t_n \in \{0,1\}$. Distribusi *bernoulli* dapat diadopsi untuk peluang $p(t|x)$ dalam *framework* probabilistik karena hanya dua nilai kelas yang mungkin, yaitu 0 dan 1 [15]. Dalam kasus klasifikasi, model prediksi mengambil bentuk kombinasi linear dari fungsi basis/kernel yang diubah oleh fungsi *logistic sigmoid* [22].

$$y(x, w) = \sigma(x; w) = \sigma(w^T \varphi(x)) \quad (2.25)$$

Keterangan:

σ = fungsi *logistic sigmoid*

w = vector bobot

dimana $\sigma(\cdot)$ Merupakan fungsi *logistic sigmoid* yang didefinisikan dengan persamaan

$$\sigma(y) = \frac{1}{1 + \exp(-y)} \quad (2.26)$$

Keterangan:

σ = fungsi *logistic sigmoid*

Berdasarkan definisi dari distribusi *Bernoulli*, *likelihood* terdefiniskan sebagai berikut

$$p(t|w) = \prod_{t=1}^N \sigma\{y(x_n; w)\}^{t_n} [1 - \sigma\{y(x_n; w)\}]^{1-t_n} \quad (2.27)$$

untuk target $t_n \in \{0,1\}$.

Persamaan *likelihood* dilengkapi dengan sebuah *prior* terhadap parameter(bobot) dalam bentuk

$$p(w|a) = \prod_{t=1}^N \frac{\sqrt{a_i}}{\sqrt{2\pi}} \exp\left(-\left(\frac{a_i w_i^2}{2}\right)\right) \quad (2.28)$$

Keterangan:

a = *hyperparameter*

dimana $a = (a_1, a_2, a_3, \dots, a_n)^T$ merupakan *hyperparameter* yang diperkenalkan untuk mengontrol kekuatan dari *prior* terhadap parameter bobot yang diasosiasikannya.

Untuk nilai α tertentu, distribusi *posterior* bobot terhadap data dapat dihitung menggunakan aturan *Bayes*, dengan persamaan

$$p(w|t, a) = \frac{p(t|w)p(w|a)}{p(t|a)} \quad (2.29)$$

Keterangan:

$p(t|w) = \text{likelihood}$

$p(w|a) = \text{prior}$

$p(t|a) = \text{evidence}$

Parameter bobot model tidak dapat diperoleh dengan cara analitik, sehingga aproksimasi *Laplacian* digunakan. Sejak $p(w|t, a)$ secara linear proposional untuk $p(t|w) \times p(w|a)$, dapat dimungkinkan untuk mencari maksimum dari persamaan.

$$\begin{aligned} \ln p(w|t, a) &= \ln\{p(t|w)p(w|a) + \ln p(t|a)\} \\ &= \sum_{n=1}^N \{t_n \ln y_n + (1 - t_n) \ln (1 - y_n)\} + \frac{1}{2} w^T A w \end{aligned} \quad (2.30)$$

untuk parameter bobot yang paling mungkin w' , dengan $y_n = \sigma\{y(x_n; w)\}^{t_n}$ dan $A = \text{diag}(a_1, a_2, \dots, a_n)$ yang terdiri dari nilai a yang diinisialisasi. Untuk mencari parameter w' pada proses reestimasi, metode *Iteratively Reweight Least-Square* dapat digunakan .

Fungsi *logistic log-likelihood* dapat diturunkan sebanyak dua kali untuk memperoleh matriks *Hessian* dengan persamaan

$$H = \nabla \nabla \ln p(w|t, a) = \varphi_n^T B \varphi_n + A \quad (2.31)$$

Keterangan :

φ = matriks kernel

B = matriks diagonal

Dimana φ merupakan matriks kernel dan $B = \text{diag}(\beta_1, \beta_2, \dots, \beta_n)$ merupakan matriks diagonal yang elemen matriksnya diperoleh dengan persamaan

$$\beta_n = \sigma\{y(x_n; w')\}\{1 - \sigma\{y(x_n; w')\}\} \quad (2.32)$$

Keterangan:

β_n = matriks diagonal

w' = parameter bobot yang akan direestimasi

Prosesre estimasi parameter bobot w didapat dengan persamaan

$$w' = w_n + \lambda \Delta \quad (2.33)$$

Keterangan:

λ = koefisien estimasi

Δ = vektor

Vektor yang diperoleh dari persamaan berikut ini

$$\Delta = H^{-1} \cdot g \quad (2.34)$$

Keterangan:

H^{-1} = invers dari matriks hessian

g = gradien negatif

dimana H^{-1} adalah invers dari matriks hessian dan g adalah gradien negatif yang diperoleh dari penurunan fungsi *logistic log-likelihood* sebanyak satu kali dengan persama

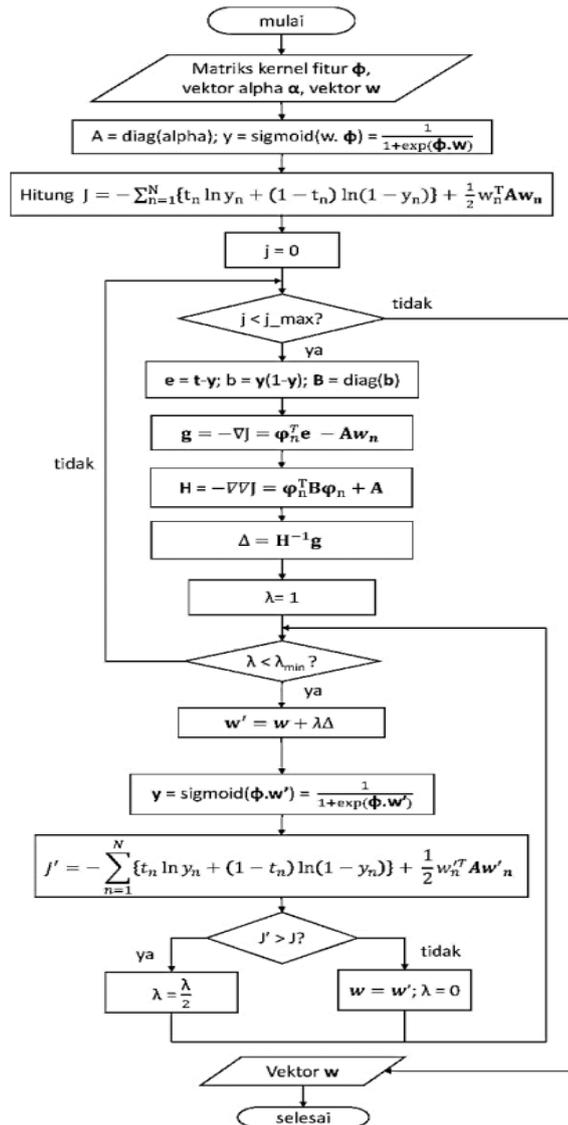
$$g = -\nabla \ln p(2|t, a) = \varphi_n^T e - Aw_n \quad (2.35)$$

Penjelasan proses estimasi parameter bobot w secara rinci dalam bentuk diagram *flowchart* dapat dilihat pada gambar 2.5. Setelah estimasi parameter bobot w dilakukan, selanjutnya *hyperparameter* α_j dilakukan reestimasi dengan menggunakan persamaan

$$a_j^{new} = \frac{y_j}{w_j^2} \quad (2.36)$$

Dimana y_j didefinisikan dengan persamaan

$$y_j = 1 - a_j \sum_{jj} \quad (2.37)$$



Gambar 2.5 Flowchart Estimasi Parameter Bobot w

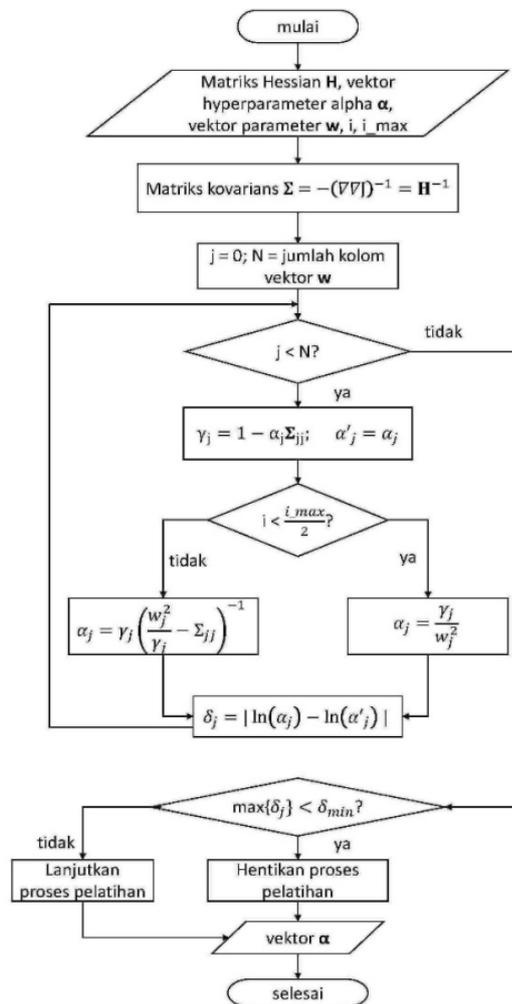
Dimana \sum_{jj} merupakan elemen diagonal matriks kovarians yang didapat dari persamaan

$$\Sigma = -(\nabla\nabla \ln p(w|t, a))^{-1} = h^{-1} \quad (2.38)$$

Dimana H^{-1} merupakan invers dari matriks *Hessian*. Selain dengan cara menggunakan persamaan (2.15), reestimasi *hyperparameter* α_j juga dapat dilakukan dengan cara “*hybrid update*” [18] dengan menggunakan persamaan

$$\begin{aligned} a_j &= \frac{Y_j}{w_j^2}; \text{ jika } i < \frac{i_{max}}{2} \\ a_j &= y_j \left(\frac{w_j^2}{y_j} - \Sigma_{jj} \right)^{-1}; \text{ jika } i \geq \frac{i_{max}}{2} \end{aligned} \quad (2.39)$$

Dimana i_{max} merupakan jumlah iterasi maksimum pada pelatihan. Penjelasan proses estimasi parameter *hyperparameter alpha* secara rinci dalam bentuk *flowchart* dapat dilihat pada gambar 2.6. Pada saat proses optimasi, banyak *hyperparameter* α_j yang akan memiliki nilai yang sangat besar, dan parameter bobot model yang berkoresponden dengan *hyperparameter* tersebut akan dipangkas (*pruned*), karena telah dalam keadaan *sparsity*. Proses optimasi terus berlanjut hingga perubahan maksimum dari α_j dibawah koefisien tertentu atau jumlah maksimum iterasi telah dicapai.



Gambar 2. 6 *Flowchart Estimasi Hyperparameter Alpha*

2.10 Alat-alat Pemodelan Sistem

Pemodelan sistem merupakan proses menyusun suatu sistem yang baru untuk menggantikan sistem yang lama secara keseluruhan atau memperbaiki sistem yang sudah ada. Berikut adalah pemodelan yang sering digunakan dalam membangun sebuah sistem.

2.10.1 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 [8].

2.10.2 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 [9].

2.10.3 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 [9].

2.11 Pengujian Confusion Matrix

Confusion matrix melakukan pengujian untuk memperkirakan obyek yang benar dan salah [18]. Urutan pengujian ditabulasikan dalam confusion matrix dimana kelas yang diprediksi ditampilkan di bagian atas matriks dan kelas yang diamati dibagian kiri. Setiap sel berisi angka yang menunjukkan berapa banyak kasus yang sebenarnya dari kelas yang diamati untuk diprediksi. [19] Model confusion matrix untuk contoh 2 kelas dapat dilihat pada Tabel 2.2 sebagai berikut

Tabel 2.2 Model Confusion Matrix

		Nilai Prediksi	
		Positif	Negatif
Nilai Aktual	Positif	TP	FP
	Negatif	FN	TN

Keterangan:

TP = jumlah nilai positive yang diklasifikasikan positif.

TN = jumlah nilai negatif yang diklasifikasikan negatif.

FP = jumlah nilai positif yang diklasifikasikan negatif.

FN = jumlah nilai negatif yang diklasifikasikan positif.

Perhitungan untuk mendapatkan akurasi dapat dilihat pada persamaan 1 sebagai berikut

$$Akurasi = \frac{TP+TN}{TP+TN+FP+FN} \quad (2.40)$$