

BAB 2

LANDASAN TEORI

2.1 Grafologi

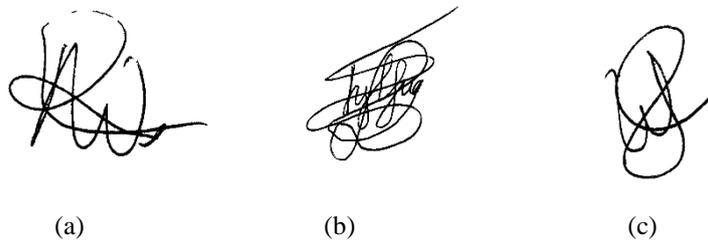
Grafologi merupakan bagian ilmu dari psikologi yang mengukur kepribadian seseorang dengan mempelajari dan menganalisis tulisan tangan ataupun tanda tangan. dengan teori grafologi, teori ini mengidentifikasi kualitas, sikap, sentimen, yang representasinya ditujukan dalam tulisan tangan. Tulisan tangan atau tanda tangan yang dapat dianalisis dalam ilmu grafologi antara lain ukuran (besar dan kecil), kemiringan (kana-kiri) tulisan atau tanda tangan [1][2][3][4]. Grafologi menjadi faktor penting dalam menganalisis karakteristik seseorang, seperti kata-kata, bahasa tubuh, bahkan foto yang menjadi penentu dalam analisis kepribadian. Grafologi dapat dikatakan sebagai karya tulis yang memiliki makna dalam mengatasi pikiran dan tubuh [11]. Grafologi memiliki dua metode untuk menganalisis tulisan tangan pertama dengan menggunakan metode jerman, bekerja dengan cara melihat secara keseluruhan tulisan seseorang, kedua metode Perancis cenderung menganalisis per huruf kemudian digabungkan. Menurut riset, keakuratan analisis grafologi mencapai 80% sampai 90% [4].

Grafologi diterapkan ke banyak bidang saat ini. Salah satu dari bidang itu analisis kepribadian, aspek intelektual perilaku dan emosional yang terkait pada orang dewasa, psiko-emosional, bimbingan akademik dan kejuruan, konseling bisnis, tulisan tangan, tanda tangan, dan pendapat ahli di pengadilan. Ahli grafologi telah terlibat dalam proses rekrutmen di banyak negara dan dalam pemilihan sumber daya manusia[11]. Saat ini ilmu grafologi telah menarik banyak minat peneliti dalam melakukan pembangunan sistem berbasis komputer. Pada perkembangannya identifikasi kepribadian dengan tanda tangan dapat dikenali dengan memanfaatkan metode machine learning. Grafologi dalam tulisan tanda tangan memiliki 9 tipe dan 15 ciri atau fitur yang menjadi dasar identifikasi kepribadian,yaitu kurva awal, coretan akhir, cangkang, coretan tengah, garis bawah, margin ekstrim, struktur titik, tanda tangan terpisah, garis terpisah

[1][12][13]. Fitur memiliki kepribadian yang unik, penjelasannya sebagai berikut.

1. Awal Kurva

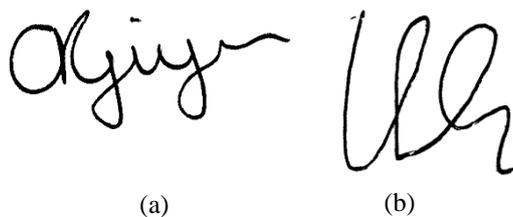
Tipe awal kurva yang memiliki 3 ciri atau fitur lengkung mundur mencerminkan karakter seseorang yang memiliki kepribadian nyaman akan masa lalu, untuk ciri lengkung tajam mencerminkan karakter seseorang yang mampu memformulasikan pikiran secara tajam dan untuk ciri lengkung lembut mencerminkan kepribadian seseorang selalu berhati-hati, ramah dan diplomatis. Contoh tanda tangan awal kurva dapat dilihat pada gambar 2.1.



Gambar 2.1 (a) Lengkung Mundur, (b) Lengkung Tajam, (c) Lengkung Lembut

2. Coretan Akhir

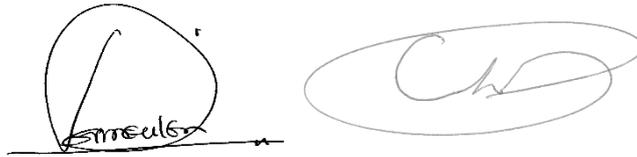
Tipe coretan akhir memiliki 2 ciri atau fitur menaik mencerminkan kepribadian seseorang sangat terbuka, mempunyai pandangan jauh kedepan, selalu berpikiran majau, dan percaya diri terhadap apa yang dilakukannya, sedangkan untuk ciri menurun mencerminkan kepribadian seseorang kurang bersemangat dalam melakukan sesuatu, selalu berfikir realistis, pesimis dan mudah putus asa. Contoh tanda tangan coretan akhir dapat dilihat pada gambar 2.2.



Gambar 2.2 (a) Menaik, (b) Menurun

3. Cangkang

Tipe Cangkang hanya memiliki 1 ciri atau fitur lengkung tertutup menggambarkan karakter seseorang memiliki ketakutan yang sangat berlebihan, introvert, tidak peduli akan lingkungan disekitarnya, kurang suka untuk bergaul atau bersosial, dan sangat sulit jika harus bekerja sama. Contoh tanda tangan cangkang dapat dilihat pada gambar 2.3.



Gambar 2.3 Cangkang

4. Coretan Tengah

Tipe coretan ditengah memiliki ciri atau fitur tunggal adanya coretan di tengah pada tanda tangan mencerminkan kepribadian seseorang memiliki rasa memiliki yang tinggi akan sesuatu, baik itu berupa barang atau lainnya yang dapat dimiliki atau sering disebut posesif. Contoh tanda tangan coretan tengah dapat dilihat pada gambar 2.4.



Gambar 2.4 Coretan Tengah

5. Garis Bawah

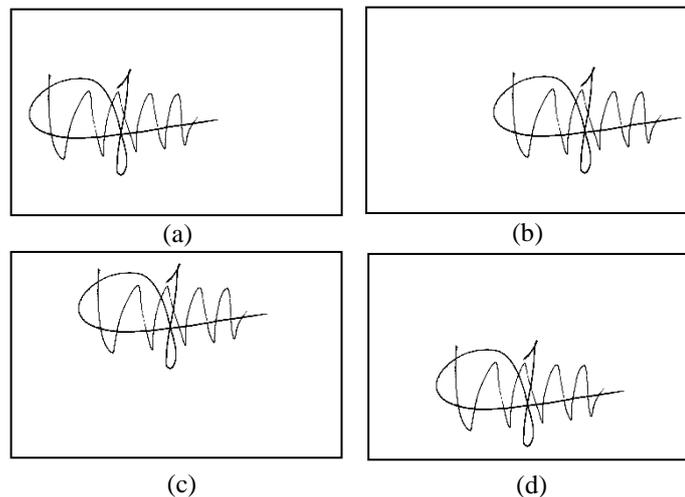
Tipe garis bawah memiliki ciri atau fitur tunggal adanya garis bawah pada tanda tangan mencerminkan karakter seseorang memiliki ide dan cara berfikir yang tidak biasa atau unik, selalu membutuhkan masukan dan dukungan dalam mengambil suatu keputusan, serta memiliki jiwa kepemimpinan yang handal. Contoh tanda tangan garis bawah dapat dilihat pada gambar 2.5.



Gambar 2.5 Garis bawah

6. Margin Ekstrim

Tipe margin ekstrim memiliki 4 ciri atau fitur margin cenderung ke kanan mencerminkan karakter seseorang selalu ceroboh dalam melakukan apapun, serta kurang perhatian, margin cenderung ke kiri mencerminkan karakter seseorang takut akan kegagalan, takut pada orang lain yang tidak dikenal, serta selalu pesimis, untuk margin cenderung di atas mencerminkan karakter seseorang selalu respek pada diri sendiri, serta selalu menjadi pribadi yang menyenangkan dan bahagia, sedangkan untuk margin cenderung di bawah mencerminkan kepribadian seseorang terlihat depresi, pribadi yang pemalu, serta selalu merasa asing akan hal yang baru atau lama. Contoh tanda tangan garis bawah dapat dilihat pada gambar 2.6.



Gambar 2.6 (a) Margin Kiri, (b) Margin Kanan, (c) Margin Atas, (d) Margin Bawah

7. Struktur Titik

Tipe struktur titik memiliki ciri atau fitur tunggal adanya titik pada tanda tangan yang mencerminkan kepribadian seseorang memiliki pendirian yang

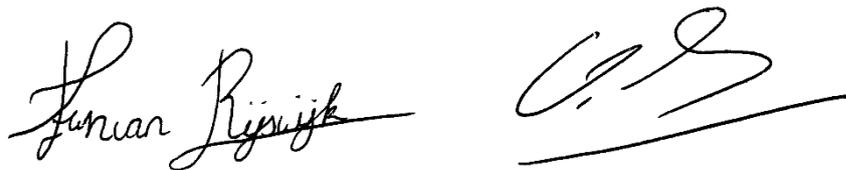
stabil, selalu menaruh rasa curiga akan hal yang tidak diketahuinya, serta selalu menjaga jarak dan tidak mudah percaya akan hal yang belum diketahui sebelumnya. Contoh tanda tangan struktur titik dapat dilihat pada gambar 2.7.



Gambar 2.7 Struktur Titik

8. Tanda Tangan Terpisah

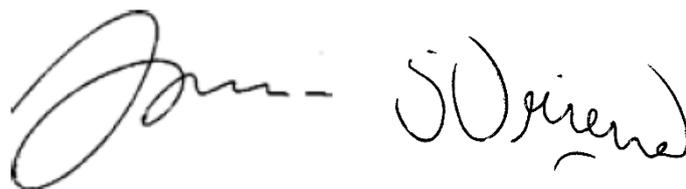
Tipe tanda tangan terpisah memiliki ciri atau fitur tunggal tanda tangan yang terpisah-pisah mencerminkan karakter seseorang memiliki pengalaman yang kurang menyenangkan pada masa lalu. Contoh tanda tangan terpisah dapat dilihat pada gambar 2.8.



Gambar 2.8 Tanda Tangan Terpisah

9. Garis Terpisah

Tipe garis terpisah memiliki ciri atau fitur tunggal adanya coretan akhir garis terputus yang mencerminkan kepribadian seseorang selalu mempunyai batasan untuk setiap keinginannya, tidak berani mengambil resiko, sering patah semangat dan ragu akan setiap keputusan yang diambilnya. Contoh tanda tangan garis terpisah dapat dilihat pada gambar 2.9.



Gambar 2.9 Garis Terpisah

2.2 Kepribadian

Kepribadian merupakan kumpulan sifat manusia untuk menggambarkan kepribadian individu/ seseorang. Kepribadian menjelaskan cara-cara terpenting di mana individu berbeda dalam hal emosi, interpersonal, pengalaman dan gaya motivasi. Kepribadian dapat dianggap sebagai profil pengguna, yang independen terhadap konteks (yang dilakukannya tidak berubah dengan waktu, lokasi, atau konteks lainnya). Saat ini, Five Factor Model of personality (FFM dianggap sebagai salah satu dari yang paling komprehensif dan merupakan model kepribadian yang paling banyak digunakan dalam merekomendasikan sistem FFM kadang-kadang disebut juga sebagai model kepribadian Big-Five (Big5)[14]. Model Big-Five sering dikenal dengan singkatan OCEAN yang terdiri dari keterbukaan, berhati-hati, kenyamanan, kebaikan, dan kestabilan [1].

1. Keterbukaan (Openness)

Openness (O), sering disebut dengan Keterbukaan, menggambarkan perbedaan antara imajinatif, orang kreatif dan sederhana, konvensional orang-orang. Golongan O tinggi biasanya individualistis, tidak sesuai dan sangat sadar akan perasaan mereka. Mereka dapat dengan mudah berpikir dalam abstraksi. Orang dengan Golongan O rendah nilai-nilai cenderung memiliki kepentingan bersama. Mereka lebih suka yang sederhana dan tidak menyukai pemikiran rumit, ambigu dan halus. Sub-faktor adalah imajinasi, minat artistik, emosi, petualangan, kecerdasan dan liberalisme.

2. Berhati-hati (Conscientiousness)

Conscientiousness (C) menyangkut cara kita mengendalikan, mengatur, dan mengarahkan impuls kita. Orang-orang dengan nilai C tinggi cenderung lebih berhati-hati sementara orang-orang dengan nilai rendah cenderung impulsif. Sub-faktor adalah self-efficacy, keteraturan, ketaatan, pencapaian prestasi, disiplin diri dan kehati-hatian.

3. Kenyamanan (Ektraversi)

Extraversion (E) memberi tahu tingkat keterlibatan dengan dunia eksternal (dalam hal ini nilai tinggi) atau kurang dari itu (nilai rendah). Sub-faktor E adalah keramahan, kegembiraan, ketegasan, tingkat aktivitas, pencarian kegembiraan dan

keceriaan. Orang ekstrovert (skor tinggi pada faktor E) cenderung bereaksi dengan antusias dan sering memiliki emosi positif ketika orang introvert (skor rendah pada faktor E) cenderung pendiam, rendah diri dan tidak terlibat dalam interaksi sosial.

4. Kebaikan (Agreeableness)

Agreeableness (A) mencerminkan perbedaan individu terkait dengan kerjasama dan harmoni sosial. Sub-domain dari faktor A adalah kepercayaan, moralitas, altruisme, kerja sama, kesederhanaan dan simpati.

5. Kestabilan (Neuroticism)

Neuroticism (N) mengacu pada kecenderungan mengalami perasaan negatif. Orang-orang dengan nilai N tinggi reaktif secara emosional. Mereka cenderung merespons secara emosional rangsangan yang relatif netral. Mereka sering berada dalam suasana hati yang buruk, yang sangat mempengaruhi pemikiran dan pengambilan keputusan mereka. Golongan skor N rendah tenang, stabil secara emosi dan bebas dari suasana hati yang buruk. Sub-faktornya adalah kecemasan, kemarahan, depresi, kesadaran diri, imoderasi dan kerentanan. Faktor neurotisme kadang-kadang disebut emosional stabilitas.

2.3 Pengenalan Pola (Pattern Recognition)

Pattern recognition merupakan salah satu algoritma *machine learning* yang berfokus pada pengenalan pola dalam dataset dan regularitas data. Sistem pengenalan pola dilatih dari data training yang telah dilabeli (*supervised learning*), namun saat ada data yang belum di labeli maka algoritma akan mengenali pola tersebut (*unsupervised learning*)[15][16]. *Pattern recognition* memiliki beberapa proses yaitu, *image retrieval*, *pre-processing* (untuk menghilangkan noise ataupun normalisasi gambar), *feature extraction*, dan *classification*. *Pattern recognition* bergerak pada proses *classification* [15] .

Pengenalan pola tanda tangan merupakan salah satu bidang pengenalan pola (*pattern recognition*) yang cukup berkembang dewasa ini, dimana aplikasinya dapat diterapkan dalam berbagai bidang khususnya dalam bidang keamanan (*security system*) seperti izin penarikan uang di bank, validasi cek dan sebagainya.

Teknologi identifikasi untuk pengenalan pola tanda tangan termasuk di dalam biometrika yang menggunakan karakteristik perilaku alami manusia [17]. Secara umum, algoritma pattern recognition bertujuan untuk menyediakan jawaban yang paling mendekati input. Hal ini bertentangan dengan algoritma pattern matching dimana mencari jawaban yang benar-benar sesuai dengan input oleh pola yang telah ada.

2.4 Pengolahan Citra Digital

Citra atau image adalah objek dua dimensi yang memiliki suatu cahaya ditinjau dari sudut pandang matematis. Sumber cahaya menerangi objek kemudian objek memantulkan kembali sebagian dari berkas cahaya tersebut, pantulan cahaya pada suatu citra kontinu dapat ditangkap oleh mata pada manusia [18]. Citra digital merupakan sekumpulan nilai-nilai real maupun kompleks yang direpresentasikan dengan deretan bit tertentu [19]. Pengolahan citra digital merupakan suatu kegiatan memanipulasi citra untuk diketahui informasi yang terkandung didalam citra[20]. Secara umum pengolahan citra digital adalah tata cara atau prosedur dalam memproses citra, baik itu memanipulasi ataupun memodifikasi citra sesuai kebutuhan masalah yang dihadapi.

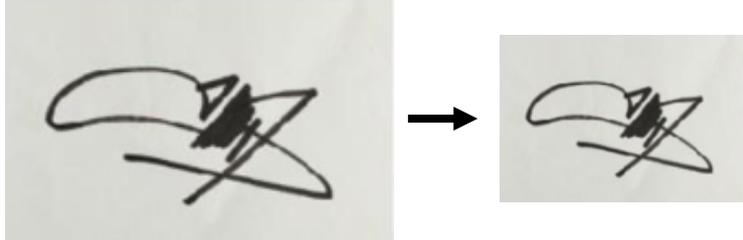
2.5 Image Pre-Processing

Pre-processing adalah proses awal yang dilakukan pada citra digital yang menjadi masukan sebelum dilakukan pemrosesan citra digital selanjutnya. Pre-processing berfokus pada peningkatan kualitas citra masukan yang diperoleh untuk menghilangkan noise-noise yang mungkin terdapat pada citra digital pada saat perekaman citra[21]. Tahap-tahap pre-processing bergantung pada masalah yang sedang dihadapi .

2.5.1 Resize

Resize merupakan mengubah ukuran gambar citra asli yang mempertimbangkan kualitas citra untuk kebutuhan penelitian. Untuk mempertimbangkan maka setiap piksel gambar berada pada interval $[0, 1]$, di mana '1' mewakili yang paling penting wilayah [22]. Terdapat beberapa teknik resize itu sendiri diantaranya scaling, cropping, seam carving, warping, dan multi-

operator. Contoh gambar masukan citra yang di resize dapat dilihat pada gambar 2.10.



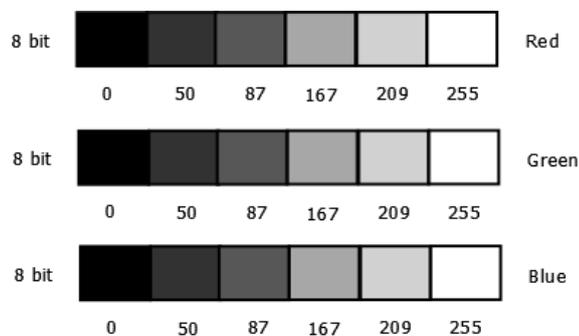
Gambar 2.10 Resize

2.5.2 Grayscale

Grayscale adalah sebuah teknik mengubah gambar (image) atau citra berwarna RGB menjadi citra berwarna abu-abu[4]. Keluaran proses greyscale merupakan sebuah citra yang hanya memiliki satu buah kanal warna pada setiap pixelnya, dengan kata lain, warna dari merah, hijau dan biru memiliki warna yang sama, yaitu hitam, keabuan dan putih. Nilai tersebut digunakan untuk menunjukan tingkat intensitas warna[19][20].

Citra grayscale memiliki warna yang lebih banyak dibandingkan citra biner dikarenakan ada nilai di antara nilai minimum (0) dan nilai maksimumnya. Banyaknya kemungkinan warna juga bergantung pada jumlah bit yang digunakan.

Citra greyscale memiliki dua warna, warna keabuan hitam untuk minimum dan warna putih untuk maksimum. Pada gambar 2.11 dijelaskan sebuah skala keabuan 8 bit jumlah kemungkinan nilainya adalah $2^8 = 256$ warna, dan nilai maksimumnya adalah $2^8 - 1 = 255$, kemungkinan warna minimal (0) dan maksimal (255). Pallette Grayscale dapat dilihat pada gambar 2.11.



Gambar 2.11 Palette Grayscale 8 bit

Untuk contoh gambar konversi masukan ke citra grayscale dapat dilihat pada gambar 2.12. Persamaan untuk mengkonversi citra RGB ke dalam citra greyscale adalah :

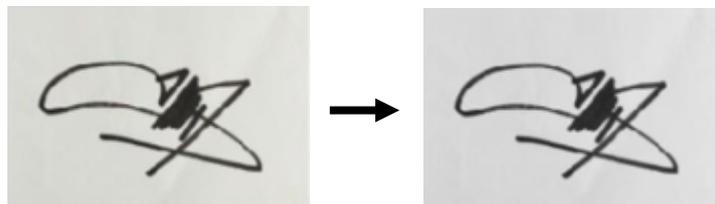
$$f(x,y) = (0.2989 * R) + (0.5870 * G) + (0.1141 * B)$$

Keterangan :

R : Komponen nilai merah (Red) dari sutau titik piksel

G : Komponen nilai hijau (Green) dari sutau titik piksel

B : Komponen nilai biru (Blue) dari sutau titik piksel



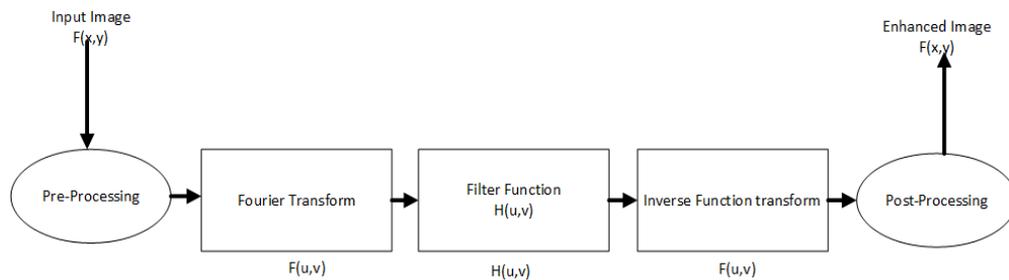
Gambar 2.12 Konversi Citra Asli Ke Grayscale

2.5.3 Perbaikan Kualitas Citra (Image Enhancement)

Beberapa gambar digital menjadi kabur disebabkan oleh noise, sehingga gambar harus ditingkatkan untuk proses lebih lanjut dan citra akan mudah untuk dikenali. Perbaikan kualitas citra (Image Enhancement) adalah meningkatkan gambar dengan mengurangi noise dan lebih mempertajamnya untuk mempermudah dalam mengenali objek. Filter ini untuk mengurangi teknik peningkatan gambar digital yang terbiasa lebih tajam pada gambar dan untuk mengurangi noise pada gambar[23]. Proses-proses yang termasuk kedalam perbaikan kualitas citra adalah perubahan kecerahan gambar (image brightening), citra negatif (image negatives), peregangan kontras (contrast stretching), perubahan histogram citra, pelembutan citra (image smoothing), penajaman (sharpening) tepi (edge), pewarnaan semu (pseudocolouring), dan lain sebagainya.

Sebagian besar pengenalan pola dan sistem pencocokan yang ada membutuhkan otomatisasi yang tangguh dalam identifikasi dan mekanisme verifikasi yang mengandalkan akuisisi gambar yang akurat. Masalah kualitas yang buruk mengharuskan dimasukkannya langkah preprocessing peningkatan sebagai

pendahuluan untuk ekstraksi fitur kualitas [24]. Ada dua jenis teknik peningkatan yang disebut teknik domain spasial dan teknik domain frekuensi yang dikategorikan lagi untuk menghaluskan dan mempertajam gambar[23]. Operasi spasial dalam pengolahan citra digital dilakukan melalui penggunaan suatu kernel konvolusi 2 dimensi. Metode image enhancement dalam operasi spasial antara lain low-pass filtering dan high-pass filtering. Operasi image Image Enhancement dapat dilihat pada gambar 2.13.



Gambar 2.13 Frekuensi Operasi Filtering

a. Low Pass-Filtering

Low-pass filtering adalah proses filter yang melewatkan komponen citra dengan nilai intensitas yang rendah dan meredam komponen citra dengan nilai intensitas yang tinggi. Low pass filter akan menyebabkan citra menjadi lebih halus dan lebih blur. Aturan kernel untuk low-pass filter adalah Semua koefisien kernel harus positif dan jumlah semua koefisien kernel harus sama dengan 1 [23].

Fungsi Low Pass-Filtering :

$$H(u, v) = \begin{cases} 1 & \text{if } D(u, v) \leq D_0 \\ 0 & \text{if } D(u, v) > D_0 \end{cases}$$

di mana $D(u, v)$: jarak dari titik (u, v) ke pusat frekuensi.

$$D(u, v) = \left[\left(u - \frac{M}{2} \right)^2 + \left(v - \frac{M}{2} \right)^2 \right]^{\frac{1}{2}}$$

Butterworth Lowpass Filter: Desain filter-domain frekuensi untuk menghilangkan noise frekuensi tinggi dengan banyaknya n .

$$H(u, v) = \frac{1}{1 + \left[\frac{D(u, v)}{D_0} \right]^{2n}}$$

Gaussian Lowpass Filters: Fungsi filter lowpass Gaussian didefinisikan sebagai:

$$H(u, v) = e^{-D^2(u, v)/2D_0^2}$$

Contoh Kernel Low Pass-Filter 3x3 :

$$H = \frac{1}{12} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 4 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

b. High Pass-Filtering

Berkebalikan dengan low-pass filtering, high-pass filtering adalah proses filter yang melewatkan komponen citra dengan nilai intensitas yang tinggi dan meredam komponen citra dengan nilai intensitas yang rendah. High pass filter akan menyebabkan tepi objek tampak lebih tajam dibandingkan sekitarnya. Aturan kernel untuk high-pass filter adalah Koefisien kernel boleh positif, negative, atau nol dan jumlah semua koefisien kernel adalah 0 atau 1 [23].

Bentuk standar filter seperti Ideal, Butterworth dan filter Gaussian. Semua filter highpass (Hhp) sering diwakili dengan filter lowpass (Hlp):

$$Hhp = 1 - hlp$$

Contoh Kernel High Pass-Filter 3x3 :

$$H = \frac{1}{9} \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

2.5.4 Segmentasi Citra

Segmentasi citra adalah pembagian daerah citra menjadi bagian bagian daerah yang lebih kecil berdasarkan letak piksel dan intensitasnya yang masih berdekatan [25][20]. Dalam pendekatan segmentasi citra terdapat dua teknik segmentasi citra yaitu pendekatan diskontinuitas dan similaritas. Pendekatan diskontinuitas beroperasi dengan cara membagi citra berdasarkan perubahan intensitas yang tiba-

tiba seperti deteksi titik, deteksi garis, dan deteksi tepi. Sedangkan pendekatan similaritas beroperasi dengan cara memecah citra ke dalam daerah yang sama menurut beberapa kriteria yang sudah ditentukan, seperti proses thresholding, region growing [20]. Segmentasi citra akan berhenti apabila proses mengenali objek awal sudah terpenuhi, dengan segmentasi citra daerah-daerah yang tidak menjadi fokus penelitian dapat diabaikan dengan berfokus pada daerah yang akan diteliti.

2.5.4.1 Edge Detection

Deteksi tepi (Edge Detection) pada suatu citra merupakan suatu proses yang menghasilkan tepi-tepi dari objek-objek citra [21][20]. Tujuan dilakukan deteksi tepi untuk menemukan bagian yang menjadi detail citra dan sekaligus untuk memperbaiki detail citra yang kabur pada saat akuisisi citra. Suatu titik (x,y) dikatakan sebagai tepi dari suatu citra jika titik tersebut mempunyai perbedaan yang tinggi dengan tetangganya [21][26]. Tepi didefinisikan sebagai perubahan nilai intensitas derajat keabuan yang mendadak (besar) dalam jarak yang singkat [20]. Tepi citra yang dihasilkan mengandung informasi penting berupa bentuk maupun ukuran objek. Edge detection ini digunakan untuk segmentasi citra dan ekstraksi data di bidang pemrosesan citra digital, metode deteksi tepi diantaranya sobel, robert, lapcian, dan canny .

1. Canny Edege Detection

Canny edge detection merupakan salah satu dari sekian banyak metode deteksi tepi yang dapat digunakan. Metode ini memiliki tiga kriteria yakni memberikan tingkat kesalahan yang sekecil mungkin, melokalisasi titik-titik tepi (jarak piksel-piksel tepi yang ditemukan deteksi dan tepi yang sesungguhnya sangat pendek), dan hanya memberikan satu respon untuk satu tepi [20][26]. Terdapat tahapan yang harus dilakukan untuk mengimplementasikan metode ini kedalam citra, berikut tahapan yang harus dilakukan Smoothing, Finding Gradient, Non-Maximum Suppresssion, Double Thresholding, Edge Tracking By Hysteresis.

a. Smoothing

Smoothing adalah mengaburkan atau menghaluskan citra masukan dari noise-noise pada saat pengambilan citra, untuk melakukan smoothing digunakan *Gaussian Filter*[26]. Persamaan Gaussian 2D dinyatakan sebagai berikut :

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Contoh Standar deviasi dengan $\sigma = 1.4$ seperti dibawah ini :

$$B = \frac{1}{159} \begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 1 & 9 & 12 & 9 & 4 \\ 2 & 4 & 4 & 4 & 2 \end{bmatrix}$$

b. Finding Gradien

Finding Gradient adalah memperkirakan gradien pada arah x dan y dengan menggunakan salah satu operator sobel, Roberts, dan Prewitt[26]. Operator sobel seperti dibawah :

$$K_{GX} = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

$$K_{GY} = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 2 \\ -1 & -2 & -1 \end{bmatrix}$$

Magnitude gradien dapat ditentukan sebagai euclidean yang diukur menggunakan pythagoras. Tujuan menguatkan tepi untuk mengurangi kompleksitas komputasi[26]. Persamaan pythagoras seperti dibawah :

$$|G| = \sqrt{G_x^2 + G_y^2}$$

$$|G| = \sqrt{G_x^2 + G_y^2}$$

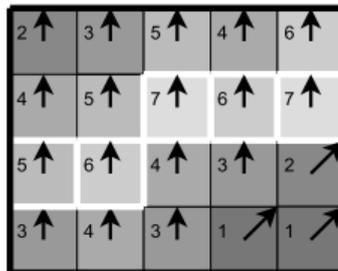
Dimana, G_x dan G_y adalah gradien pada masing-masing arah x dan y.

Untuk menentukan tepian, arah tepian sudut harus ditentukan dan disimpan seperti ditunjukkan dalam persamaan dibawah ini.

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

c. Non-Maximum Supression

Non-Maximum Supression merupakan perampingan garis tepi yang muncul dengan menerapkan nonmaximum suppression sehingga menghasilkan garis tepian yang ramping. Non-Maximum Suppression dilaksanakan dengan memperhatikan dua titik tetangga yang terletak pada arah tepi. Jika nilai pixel titik perhatian lebih besar daripada gradien kedua tetangga, maka simpan nilai tepian tersebut. Sebaliknya, jika pixel titik perhatian lebih kecil daripada nilai salah satu atau kedua gradien tetangga, nilainya akan diubah menjadi 0 atau dihapus[20]. Ilustrasi penghapusan seperti gambar 2.14 .



Gambar 2.14 Ilustrasi Penghapusan No-Maximum Supression

d. Double Thresholding

Ambang batas ganda (Double Thresholding) bertujuan mengklasifikasi dua buah nilai High-threshold (T_2) dan Low-threshold (T_1), dengan $T_2 \approx 2T_1$. Jika nilai pixel lebih besar atau sama dengan T_2 maka diatur nilai 255 atau menjadi titik kuat, jika nilai pixel kurang dari atau sama dengan T_1 maka diatur menjadi 0 atau dihapus. Pixel diantara T_1 dan T_2 disebut kandidat pixel tepi atau tepian yang lemah maka sementara diberi nilai 128[20][26].

e. Edge Tracking By Hysteresis

Edge tracking by hysteresis bertujuan memperoleh tepian final dengan menekan semua sisi yang tidak terhubung pada tepian yang sangat kuat. Tepian yang setidaknya mengandung 1 piksel tepian yang kuat akan dipertahankan, selain itu akan di ubah menjadi 0 atau dihapus[20][26]. Nilai 128 selanjutnya dilakukan pengecekan pada pixel dari 8 arah tetangganya, sehingga pixel hanya bernilai 0 atau 255. Perubahan nilai 128 menjadi nilai 255 apabila semua kondisi terpenuhi

yakni jika salah satu atau semua pixel pada 8 arah tetangganya bernilai 255. Proses pengujian dilakukan sampai tidak ada lagi perubahan dari nilai 128 menjadi 255[20].

2.5.4.2 Thresholding

Thresholding merupakan salah satu metode dari sekian banyaknya segmentasi citra di mana prosesnya mengubah derajat keabuan citra yang bergantung pada suatu nilai batas atau yang disebut nilai threshold [20]. Thresholding merupakan teknik penting dalam segmentasi image. Thresholding diklasifikasikan menjadi dua, Global Thresholding dan Local Thresholding, Global thresholding dibagi menjadi Tradisional, Iteratif, Multistage [27].

Metode thresholding bertujuan memisahkan suatu citra kedalam dua bagian yaitu objek dan latar belakang. Jika bagian piksel pada citra keabuan lebih besar atau sama dengan nilai threshold (T) akan diubah menjadi putih (1) sebaliknya jika bagian piksel pada citra keabuan lebih kecil dari nilai threshold (T) akan diubah menjadi hitam (0). Output dari metode ini yaitu sebuah bilangan biner yang hanya mengenali intensitas 0 dan 1 atau putih dan hitam [20][27]. persamaan untuk melakukan thresholding adalah :

$$g(x,y) = \begin{cases} 1, & \text{jika } f(x,y) \geq T \\ 0, & \text{jika } f(x,y) < T \end{cases}$$

Persamaan berikut menjelaskan bahwa $g(x,y)$ adalah nilai citra biner dan $f(x,y)$ adalah nilai pixel pada citra *greyscale* dan T adalah nilai batas atau threshold.

2.5.4.3 Segmentasi Vertikal dan Horizontal

Segmentasi terbagi menjadi dua yaitu *projection vertikal* yang menghitung sumbu y berdasarkan kolom dan *projection horizontal* yang menghitung sumbu x berdasarkan baris[28].

a. Projection vertikal

Menjumlahkan piksel putih yang tegak lurus dengan sumbu y, yang diwakili dengan vektor Pv dengan ukuran N yang dapat dilihat pada persamaan berikut :

$$Pv[i] = \sum_{j=1}^M S(i,j)$$

Dimana, M merupakan tinggi citra dan $Pv[i]$ jumlah piksel pada kolom i citra.

b. Projection horizontal

Menjumlahkan piksel putih yang tegak lurus dengan sumbu x, yang diwakili dengan vektor Ph dengan ukuran M yang dapat dilihat pada persamaan berikut :

$$Ph[j] = \sum_{i=1}^M S(i,j)$$

Dimana, M merupakan lebar citra dan $Ph[K]$ jumlah piksel pada baris k citra.

2.5.5 Ekstrasi Fitur

Fitur merupakan sifat unik dari suatu objek. Terdapat dua fitur, pertama fitur alami merupakan fitur bagian dari gambar, misal kecerahan objek. kedua fitur buatan merupakan fitur yang diperoleh dengan operasi tertentu pada gambar, misalnya histogram tingkat keabuan. Maka dapat disimpulkan ekstraksi fitur merupakan proses pencarian ciri pembeda yang membedakan suatu objek dari objek yang lain yang terdapat pada citra [29]. Wilayah tersebut dapat didefinisikan dalam lingkungan global atau lokal dan dibedakan oleh bentuk, tekstur, ukuran, intensitas, sifat statistik, dan sebagainya. Feature extraction dilakukan dengan cara menghitung jumlah titik atau pixels yang ditemui dalam setiap pengecekan, dimana pengecekan dilakukan dalam berbagai arah tracing pengecekan pada koordinat kartesian dari citra digital yang dianalisis, yaitu vertikal, horizontal, diagonal kanan, dan diagonal kiri.

Bentuk dari suatu objek adalah karakter konfigurasi permukaan yang diwakili oleh garis dan kontur . Fitur bentuk dikategorikan bergantung pada teknik yang digunakan. Kategori tersebut adalah berdasarkan batas boundary based dan berdasarkan daerah region based. Teknik berdasarkan batas (boundary based) menggambarkan bentuk daerah dengan menggunakan karakteristik eksternal, contohnya adalah piksel sepanjang batas objek. Sedangkan teknik berdasarkan

daerah (region based) menggambarkan bentuk wilayah dengan menggunakan karakteristik internal, contohnya adalah piksel yang berada dalam suatu wilayah.

Fitur bentuk yang biasa digunakan adalah :

- Wilayah (area) yang merupakan jumlah piksel dalam wilayah digambarkan oleh bentuk foreground;
- Lingkar (perimeter) adalah jumlah dari piksel yang berada pada batas dari bentuk . perimeter didapatkan dari hasil deteksi tepi;
- Kekompakan compactness;
- Euler number atau faktor E adalah perbedaan antara jumlah dari connected component (C) dan jumlah lubang (H) pada citra.

2.5.5.1 Hu Moment

Hu Moment digunakan sebagai fitur dalam pemrosesan citra, pengenalan bentuk untuk klasifikasi. Nilai moment invariant ini adalah invariant terhadap translasi, skala dan rotasi bentuk (*Rotation Scale Translation (RST)-invariant*)[10]. Karena itu, itu sangat berguna sebagai metode ekstraksi bentuk kontur. Kontur adalah daftar piksel yang mewakili kurva pada gambar. Moment invariant dihitung berdasarkan informasi yang diberikan oleh boundary bentuk dan daerah interiornya. Proses perhitungan nilai HuMoment (ϕ) adalah sebagai berikut.

- Menghitung momen spasial

$$m_{pq} = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} x^p y^q f(x, y)$$

M : jumlah baris/resolusi panjang citra

N : jumlah kolom/resolusi lebar citra

pq : orde momen

x, y : kordinat piksel

$f(x, y)$: intensitas piksel di titik x, y

- Menghitung nilai momen pusat

Menghitung pusat kordinat dari area atau massa y, x , sebelum melakukan pencarian momen pusat, maka dicari terlebih dahulu titik kordinat sebagai berikut.

$$\bar{x} = \frac{m_{10}}{m_{00}}, \bar{y} = \frac{m_{01}}{m_{00}}$$

- Menghitung momen pusat (μ_{pq}) orde 2 ($\mu_{11}, \mu_{20}, \mu_{02}$) dan orde 3 ($\mu_{30}, \mu_{03}, \mu_{21}, \mu_{12}$) dengan persamaan berikut:

$$\mu_{pq} = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} (x - \bar{x})^p (y - \bar{y})^q f(x, y)$$

- Menghitung normalisasi momen pusat (η_{pq}) orde 2 ($\eta_{11}, \eta_{20}, \eta_{02}$) dan orde 3 ($\eta_{30}, \eta_{03}, \eta_{21}, \eta_{12}$) dengan persamaan berikut:

$$\eta_{pq} = \frac{\mu_{pq}}{\eta_{00}^y}, y = \frac{p+q}{2} + 1$$

3. Menghitung invarian moment pusat (η_{pq}), nilai Hu momen ϕ dengan

perhitungan sebagai berikut :

$$\phi_1 = (\eta_{20}) + (\eta_{02})$$

$$\phi_2 = (\eta_{20} + \eta_{02})^2 + (\eta_{11})^2$$

$$\phi_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2$$

$$\phi_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2$$

$$\phi_5 = (\eta_{30} - 3\eta_{12}) (\eta_{30} + \eta_{12}) [(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + (3\eta_{21} - \eta_{03}) (\eta_{21} + \eta_{03}) [3(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2]$$

$$\phi_6 = (\eta_{20}) + (\eta_{02}) [(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + (4\eta_{11})(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03})$$

$$\phi_7 = (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12}) [(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + (3\eta_{12} - \eta_{30}) (\eta_{21} + \eta_{03}) [3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]$$

4. Logaritma Hu Moment

$$Hi = \text{sign}(hui) \log |hui|$$

2.6 Algoritma Smooth Support Vector Machine (SSVM)

Smooth Support Vector Machine (SSVM) pengembangan dari metode Support Vector Machine(SVM). Tujuan dari SVM maupun SSVM adalah sama yaitu menemukan fungsi pemisah (hyperplane) yang optimal sebagai pemisah dua buah kelas atau lebih pada input space. Hyperplane terbaik adalah memiliki garis yang maksimal yang memisahkan kedua kelompok. Perbedaan mendasar SVM dengan SSVM adalah pada SSVM melibatkan fungsi smoothing didalamnya [30]. Support Vector Machines (SVM) adalah suatu teknik untuk melakukan prediksi, baik dalam kasus klasifikasi maupun regresi. Pada ruang berdimensi tinggi, akan dicari hyperplane (fungsi pemisah) yang dapat memaksimalkan jarak (margin) antara dua kelas data. Klasifikasi menggunakan SVM dapat dijelaskan secara sederhana yaitu usaha untuk mendapatkan garis sebagai fungsi pemisah terbaik yang dapat memisahkan dua kelas yang berbeda (+1,-1) pada ruang input[7].

SVM memanfaatkan optimasi dengan quadratic programming, sehingga untuk data berdimensi tinggi dan data jumlah besar SVM menjadi kurang efisien. Oleh karena itu dikembangkan smoothing technique yang menggantikan plus function SVM dengan integral dari fungsi sigmoid neural network yang selanjutnya dikenal dengan Smooth Support Vector Machine (SSVM)[7].

Diberikan masalah klasifikasi dari n objek dalam ruang dimensi R^p sehingga susunan data berupa matriks A berukuran $n \times p$ dan keanggotaan tiap titik terhadap kelas $\{+1\}$ atau $\{-1\}$ atau yang didefinisikan pada diagonal matriks D berukuran $m \times n$, problem optimasinya adalah :

$$\min_{w,b,\xi} \frac{c}{2} \xi' \xi + \frac{1}{2} (w' w + b^2), \quad \text{dengan} \quad \text{kendala}$$

$$D(Aw + eb) + \xi \geq e, \xi \geq 0.$$

Solusi problem adalah :

$$\xi = (e - D(Aw + eb)),$$

Dimana ξ adalah variabel slack yang mengukur kesalahan klasifikasi. Kemudian dilakukan substitusi dan konersi, sehingga persamaan dapat ditulis sebagai berikut :

$$\min_{w,b} \frac{v}{2} \| (e - D(Aw - eb)) \|_2^2 + \frac{1}{2} (w'w + b^2),$$

Fungsi objektif dalam persamaan diatas tidak memiliki turunan kedua. Teknik smoothing yang disusulkan dilakukan dengan mengganti fungsi plus dengan $p(x,a)$ yaitu integral dari fungsi sigmoid neural network $(1 + \xi^{-ax})^{-1}$ atau sebagai berikut :

$$p(x, a) = x + \frac{1}{a} \log(1 + e^{-ax}), a > 0$$

Dimana α adalah parameter smoothing. Dengan menggantikan fungsi plus dengan $p(x,\alpha)$ maka diperoleh model SSVM sebagai berikut :

$$\min_{(w,b) \in \mathbb{R}^{p+1}} \Phi_\alpha(w, b) = \min_{(w,b) \in \mathbb{R}^{p+1}} \frac{c}{2} \| p(e - D(Aw - eb)) \|_2^2 + \frac{1}{2} (w'w + b^2),$$

secara umum, problem optimasi SSVM dapat ditulis sebagai berikut :

$$\begin{aligned} \min_{(w,b) \in \mathbb{R}^{p+1}} \Phi_\alpha(w, b) \\ = \min_{(w,y) \in \mathbb{R}^{p+1}} \frac{c}{2} \| p(e - D(k(x_i, x_j)Dw - eb), \alpha) \|_2^2 \\ + \frac{1}{2} (w'w + b^2), \end{aligned}$$

Yang diselesaikan dengan iterasi Newton armijo dan $K(x_i, x_j)$ merupakan fungsi kernel yang dalam penelitian ini digunakan gaussian atau bisa dirumuskan berikut $K(x_i, x_j) = \exp\left(-\gamma (\|x_i - x_j\|^2)\right)$ dengan parameter kernel γ .

Persamaan 1 :

$$\Phi_\alpha(w_i, b_i) - \Phi_\alpha((w_i, b_i) + \lambda_i d_i) \geq -\delta \lambda_i \nabla \Phi_\alpha(w_i, b_i) d_i$$

Persamaan 2 :

$$(w_{i+1}, b_{i+1}) = (w_i, b_i + \lambda_i d_i)$$

Saat iterasi pada algoritma newton armijo berhenti, diperoleh nilai w dan b yang konvergen. Dengan demikian fungsi pemisah yang diperoleh untuk kasus klasifikasi linier adalah :

$$f(x) = \text{sign}(w'x + b),$$

Sedangkan fungsi pemisah untuk kasus klasifikasi nonlinier adalah sebagai berikut :

$$f(x) = \text{sign}(w'x + b) = (u'D'K(x_i, x_j) + b)$$

Perumusan program linier SVM 1-norm adalah salah satu cara untuk memilih atribut(feature selection) diantara varian norm SVM, problem tersebut sebagai berikut :

$$\min_{(w, b, s, \xi) \in \mathbb{R}^{(2p)+1+n}} C e' \xi = e' s, \quad \text{dengan kendala}$$

$$D(Aw + eb) + \xi \geq e - s \leq w \leq s \xi \geq 0$$

Solusi dari w mampu menghasilkan model yang parsimoni dan bersifat sparsity. Jika nilai dari elemen vektor $w_p = 0$, maka variabel p tidak berkontribusi dalam penentuan kelas. Kontribusi atribut atau variabel prediktor dapat dinilai dari besarnya nilai w_i untuk masing-masing atribut, dengan $l = 1, 2, \dots, p$.

2.7 Machine Learning

Pembelajaran Mesin atau Machine Learning adalah ilmu yang mempelajari bagaimana komputer memahami data, bagaimana data mempunyai hubungan, dan ilmu komputer, dengan penekanan algoritma yang efisien. Machine learning sering disebut juga Ilmu hasil dari kombinasi ilmu matematika dan komputer sains yang membangun model statistika dari kumpulan data besar [31]. Pembelajaran mesin dapat diaplikasikan untuk memecahkan berbagai masalah praktis. Tujuan algoritma pembelajaran mesin termasuk klasifikasi, regresi, pengenalan pola, data mining hingga informatika.

Pembelajaran mesin memungkinkan mesin membuat model dari input pengamatan data untuk membuat keputusan yang lebih rinci[32]. Machine learning sangat membantu dalam mengatasi masalah-masalah yang selama ini sulit dipecahkan, sehingga masalah yang rumit bisa menjadi hal yang mungkin dilakukan. Sebelum menjadi mesin pintar, machine learning harus mampu belajar dan beradaptasi dengan lingkungan yang mengharuskan sistem dapat belajar dan

menyesuaikan dengan keadaan, pengembang tidak perlu meramalkan atau memberi solusi untuk situasi yang mungkin terjadi[33]. Secara umum machine learning terbagi tiga kategori, yaitu *survised learning*, *unsurvised learning*, dan *reinforcement learning*.

2.7.1 Survised Learning

Pembelajaran mesin terawasi dimulai dengan tujuan memprediksi yang diketahui output atau target. Dalam pembelajaran mesin setiap data masukan dinilai berdasarkan kesamaan kinerja set data, masalah pembelajaran yang diawasi berulang meliputi pengenalan tulisan tangan (seperti mengenali tanda tangan), mengklasifikasikan gambar, dan klasifikasi dokumen [31][34]. Pembelajaran mesin terawasi bersifat dinamis apabila set data di ubah atau ditambah, sehingga pembelajaran mesin terawasi akan sangat bergantung kepada set data , tanpa set data pembelajaran mesin ini tidak akan bekerja dengan baik atau tidak akan memecahkan masalah dengan mendekati kebenaran[33].

2.7.2 Unsurvised Learning

Unsurvised learning atau pembelajaran mesin tidak terawasi berbanding terbalik dengan supervised learning dimana survised learning membutuhkan pelatihan, sedangkan pada unsupervised learning tidak membutuhkan pelatihan. Perbedaan utamanya adalah pada unsupervised learning tidak ada desired output. Contoh penggunaan unsupervised learning adalah metode clustering dan algoritma yang terkenal adalah K-means. Unsupervised learning adalah mencari sifat-sifat (properties) data[33][35][36].

2.7.3 Reinforcement Learning

Reinforcement Learning (RL) merupakan Paradigma baru dalam Machine Learning. Reinforcement Theory (RT) dalam ilmu psikologi sebagai dasar dalam konsep Reinforcement Learning (RL). Teori ini menjelaskan bagaimana dalam dinamika kehidupan seseorang bisa memilih, menentukan dan mengambil keputusan[33][35]. Teori ini bisa diterapkan pada situasi tertentu yang seringkali dihadapi manusia. Reinforcement Theory ini menjelaskan bagaimana perilaku manusia itu adalah hasil kompilasi dari pengalaman-pengalaman yang ia temui/hadapi sebelumnya, (Consequences influence behavior). Contoh

pemanfaatan RL adalah machine yang disebut SNARC (Stochastic Neural-Analog Reinforcement Calculator)[35].

2.8 Bahasa Pemrograman

Bahasa pemrograman diperlukan untuk menjalankan instruksi-instruksi apa yang harus dilakukan komputer. Komputer tidak bisa memahami bahasa natural manusia, sehingga diperlukan penggunaan bahasa perantara antara manusia dengan komputer untuk diimplementasikan ke dalam komputer[37]. terdapat beberapa macam bahasa pemrograman untuk implementasi perangkat lunak diantaranya Java, PHP, C++,C# dan Python.

2.8.1 Bahasa Pemrograman Python

Python adalah bahasa pemrograman tingkat tinggi yang diciptakan oleh Guido van Rossum dan dirilis pertama kali pada tahun 1991/ filosofi desain Python menekankan keterbacaan kode dengan penggunaan spasi - spasi yang signifikan. Konstruksi bahasanya dan pendekatan berorientasi objek bertujuan untuk membantu programmer menulis kode yang jelas dan logis. Dengan desainnya, python memiliki kode yang mudah dibaca dan dimengerti [38].

Tools dan teknik pembelajaran mesin menjadi lazim dalam proyek perangkat lunak intensif data. Di antara berbagai bahasa digunakan untuk Ilmu Data, Python telah menjadi salah satunya bahasa yang paling populer karena banyaknya koleksi library untuk mengatur dan menganalisis data. Mining Python pada proyek akan sangat membantu untuk meningkatkan desain bahasa, perpustakaan perangkat tambahan, deteksi bug serta membuka arahan penelitian baru [39]. Beberapa kelebihan bahasa pemrograman python adalah sebuah bahasa pemrograman tingkat tinggi, mudah dipelajari, berorientasi objek, serta portable pada seluruh sistem operasi dan library yang sangat banyak menjadikan pengerjaan pengkodean lebih efektif dan efisien.

2.8.2 Open CV

Open CV merupakan library atau pustaka dari bahasa pemrograman python, Open CV pustaka yang menangani pengelolaan citra pada komputer. Pendukung seperti Numpy perpustakaan yang dimanfaatkan untuk perhitungan numerik pada

citra. Open digunakan untuk memudahkan dalam pengimplementasian algoritma kedalam kode sistem[32]. Selain itu, OpenCV mendukung berbagai macam pemrograman bahasa seperti C ++, Python, Java, dll., dan tersedia di berbagai platform termasuk Windows, Linux, OS X, Android, dan iOS. OpenCV-Python adalah "Python Application Programming Interface (API) untuk OpenCV yang menggabungkan bahasa pemrograman Python dengan C ++ API. "

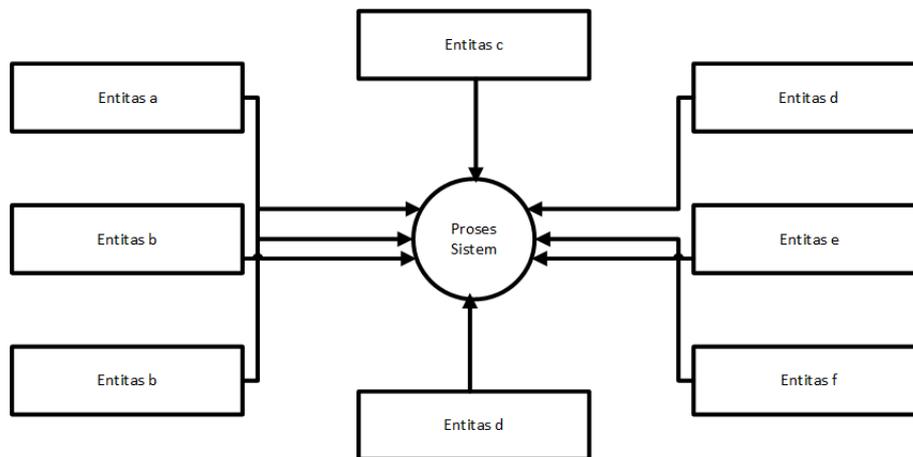
2.9 Model Proses

Model proses merupakan penataan dan pendokumentasian proses, input, output dan simpanan data dalam sistem. Model proses sebagai alat untuk merekayasa perangkat lunak untuk memodelkan hal lain yang bukan perangkat lunak. Terdapat alasan kuat dilakukan pemodelan diantaranya terlibat pada hal penting meskipun sistem belum selesai dibuat, mendiskusikan perubahan dan biaya dan kebutuhan, dan menguji penganalisa sistem terhadap kebutuhan membantu desaigner sistem membangun sistem.

2.9.1 Digram konteks

Diagram konteks merupakan penggambaran sistem yang akan dibangun secara keseluruhan, semua entitas harus digambarkan sedemikian rupa, sehingga terlihat data yang mengalir pada input-proses-output antara entitas dengan sistem[40][41]. Diagram konteks menggunakan tiga buah simbol yaitu simbol untuk melambangkan entitas, simbol untuk melambangkan alur data dan simbol untuk melambangkan proses.

Diagram konteks hanya boleh terdiri dari satu proses saja, tidak boleh lebih, dan pada diagram konteks tidak digambarkan data store. Proses pada Diagram konteks biasanya tidak diberi nomor. Contoh Digaram kontek dapat dilihat pada gambar 2.15.

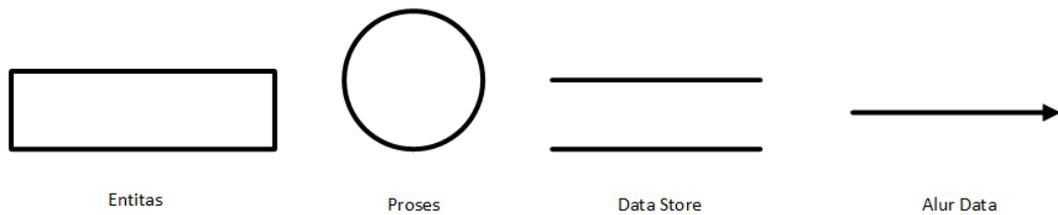


Gambar 2.15 Contoh Diagram Konteks

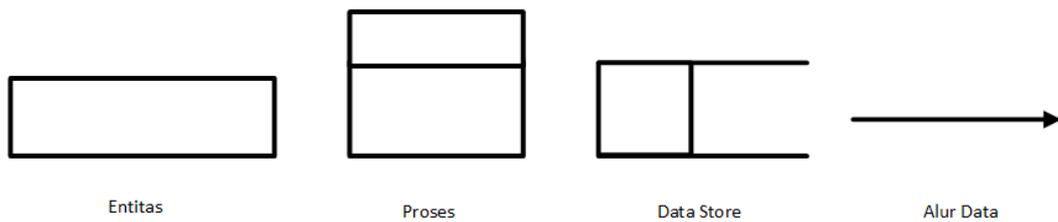
2.9.2 Data Flow Diagram(DFD)

Data Flow Diagram (DFD) adalah sebuah alat untuk merancang sistem yang berorientasi kepada alur data yang mudah untuk dikomunikasikan oleh programmer. Dalam pengembangan aplikasi DFD yang juga disebut DAD (Diagram Aliran Data), definisi DFD adalah proses data yang menggambarkan, dari mana data berasal, dari mana data keluar dari sistem, dan maka data akan disimpan. Jadi ada aliran data interaksi ke penyimpanan terakhir[42][43]. Definsi lain mengenai DFD adalah hubungan untuk menggambarkan sistem secara otomatis atau terkomputerisasi, secara manual presentasi dapat diberikan dalam bentuk sistem yang saling berhubungan sesuai dengan aturan[42].

DFD memiliki dua teknik dasar penggambaran simbol DFD yang umum dipakai, pertama adalah Gane and Sarson sedangkan yang kedua adalah Yourdon and De Marco[43]. Perbedaan yang mendasar pada teknik tersebut adalah lambang dari simbol yang digunakan. simbol menurut Yourdon and De Marco digunakan untuk menggambar DFD dapat dilihat pada gambar 2.16 dan menurut Gane and Sarson dapat dilihat pada gambar 2.17.



Gambar 2.16 Simbol Menurut Yourdan De Marco



Gambar 2.17 Simbol DFD Menurut Gane and Sarson

a. Entitas

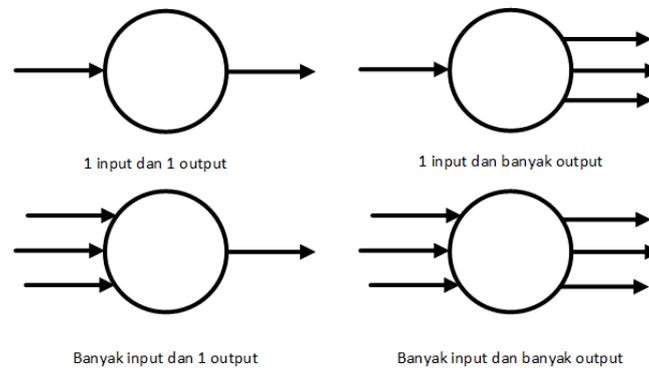
Simbol entitas menyatakan kesatuan diluar sistem (external entity) yang memberikan input ke sistem / menerima output dari sistem berupa organisasi, orang, atau sistem lain. Jenis-jenis entitas pada Data Flow Diagram (DFD) dapat dilihat pada gambar 2.18.



Gambar 2.18 Jenis-jenis Entitas

Simbol proses untuk menunjukkan suatu kegiatan sedang dilakukan. Proses tersebut harus menerima aliran data (input) dan menghasilkan data saat ini (output).

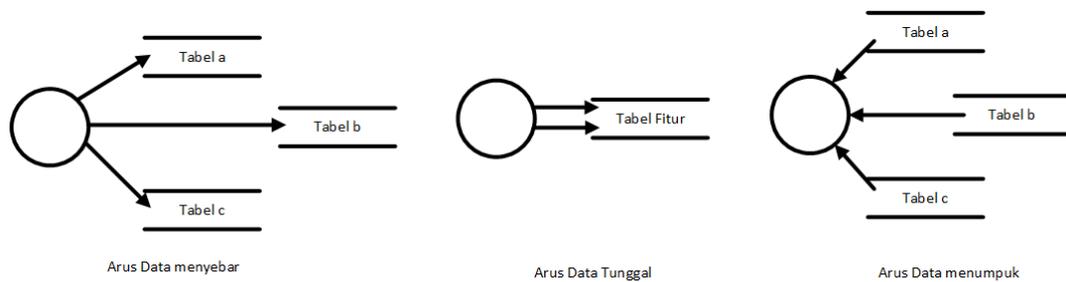
Proses mempunyai input tapi tidak menghasilkan output disebut black hole (lubang hitam). Sebaliknya, proses menghasilkan output tapi tidak menerima input disebut miracle (ajaib) [43]. Jenis-jenis proses pada Data Flow Diagram (DFD) dapat dilihat pada gambar 2.19.



Gambar 2.19 Jenis-jenis Proses

b. Alur Data

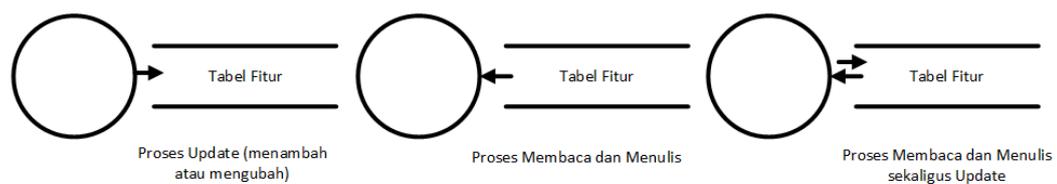
Simbol Aliran data untuk menyatakan aliran data pada sistem (antar proses, antara proses & entitas, serta antara proses & data store). Jenis-jenis arus data pada Data Flow Diagram (DFD) dapat dilihat pada gambar 2.20.



Gambar 2.20 Jenis-jenis Arus Data

c. Data Store

Untuk menunjukkan data yang disimpan ke database dari sistem komputer, biasanya berupa tabel. Selain itu, untuk menunjukkan ada database yang di baca dan di tulis yang akan dibawa ke suatu proses. Hanya proses yang bisa mengakses pada data store. Jenis-jenis data store pada Data Flow Diagram (DFD) dapat dilihat pada gambar 2.21.



Gambar 2.21 Jenis-jenis Data Store

2.10 Pengujian Perangkat Lunak

Pengujian merupakan bagian dari siklus tahapan rekayasa perangkat lunak yang bertujuan untuk menemukan kesalahan dari perangkat lunak yang telah

2.10.1 Pengujian Sistem

Pengujian sistem adalah proses mencari kesalahan program yang tidak terlihat atau tidak muncul selama proses pembuatan program. Secara umum ada 3 kasus kondisi dari hasil pengujian yaitu kondisi keliru (*fault*), kondisi salah (*error*) dan kondisi gagal (*failure*)[44]. Kondisi keliru adalah apabila terjadi kesalahan pada baris kode, apabila kondisi fault ini dieksekusi maka program akan menghasilkan respon yang error sehingga program tidak menjalankan fungsinya dengan baik. Pengujian perangkat lunak merupakan aktivitas yang melibatkan beberapa kegiatan yang saling berkaitan. Tujuan pengujian perangkat lunak adalah untuk meningkatkan rasa percaya diri atas pengembang perangkat lunak terhadap fungsi-fungsi perangkat lunak. Secara umum teknik pengujian perangkat lunak dapat diklasifikasikan ke dalam pengujian *blackbox* dan *whitebox*.

a. Blackbox Test

Pengujian *blackbox* juga disebut sebagai pengujian fungsional perangkat lunak. Dengan *blackbox*, penguji perangkat lunak seharusnya tidak memiliki akses ke kode program perangkat lunak karena Pengujian black box tidak berkaitan dengan mekanisme internal suatu sistem, ini hanya fokus pada output yang dihasilkan sebagai respons terhadap input yang dipilih dan kondisi eksekusi[45].

Function testing sangat berhubungan dengan pengujian blackbox. Teknik ini berfokus menemukan kesalahan pada fungsi yang tidak benar, kesalahan antarmuka, kesalahan basis data, kesalahan performansi, kesalahan inisialisasi dan terminasi[44][45].

b. Whitebox Test

Pengujian *whitebox* merupakan pengujian logika algoritma yang diterapkan kedalam kode program, pengujian ini digunakan untuk men-debug , menemukan typographical errors secara acak, dan menemuka asumsi-asumsi yang salah

penerapannya[45]. Pengujian ini berfokus pada kode-kode program perangkat lunak yang dilakukan oleh general programmer atau independent tester.

Structural testing mempunyai peranan penting dalam pengujian *whitebox*. Dengan teknik ini berfokus pada implementasi kode program, cabang program, spesifik program dan hal lain yang berkaitan dengan internal sistem pengkodean dan logika program[45].

2.10.2 Pengujian Akurasi

Akurasi merupakan seberapa dekat suatu angka hasil pengukuran terhadap angka sebenarnya (true value atau reference value). Tingkat akurasi diperoleh dengan persamaan sebagai berikut :

$$Akurasi = \frac{Jumlah\ Ciri\ Sama}{Jumlah\ Seluruh\ Ciri} \times 100\%$$