

BAB 2

LANDASAN TEORI

2.1 Grafologi

Grafologi merupakan pembelajaran mengenai tulisan tangan. Menurut Webster dalam Olyanova (1960), Grafologi adalah seni atau ilmu yang menarik kesimpulan mengenai karakter, kepribadian, watak, atau sikap-sikap dari tulisan tangan seseorang. Definisi lain dari Elaine menggambarkan Grafologi sebagai “tulisan otak” [4].

Untuk menulis, manusia menggunakan 3 (tiga) aspek dalam dirinya yaitu tubuh, mental, dan emosional. Saat menulis manusia sedang menuangkan gambaran tubuh, mental, dan emosionalnya ke dalam tulisan tangannya. Aspek tubuh mengungkap kondisi fisik, kekuatan dan kesehatan penulisnya. Aspek mental mengungkap kecerdasan dan kepribadian. Aspek emosional mengungkap kondisi emosi atau perasaan penulisnya.

Tulisan tangan disebut juga tulisan otak atau fikiran (*brainwriting*), dengan segala kemampuannya, dengan demikian grafologi dapat dipergunakan sebagai alat tes untuk mengungkap karakter dan kepribadian penulisnya, dalam hal ini dinamakan *graphotest*.

2.2 Sejarah Grafologi

Grafologi memiliki sejarah yang telah berumur panjang dengan latar belakang filosofis berkaitan dengan hakikat manusia dan perilaku. Pada tahun 300 M, filsuf Yunani Aristoteles telah menyampaikan pandangannya mengenai hubungan antara tulisan tangan, kata-kata, dan kepribadian [4].

Pada abad ke-2 M, C.Suetonius Tranquillus sudah mencatat keunikan dari tulisan tangan Caesar. Pada abad ke-11 M, bangsa Tiongkok juga mencatat hubungan antara tulisan tangan dan kepribadian. Francois Demelle, seorang berkebangsaan Prancis pertama kali mengenalkan teori yang menyatakan bahwa tulisan tangan merupakan perwujudan dari karakter seseorang. Pada tahun 1622,

Camilo Baldi menulis buku tentang Grafologi di Italia, Camilo Baldi adalah doktor Italia di bidang Kedokteran dan Filsafat serta professor di Universitas Bologna.

Ilmu analisis tulisan tangan, atau grafologi yang formal, muncul dari dua pastor Prancis pada abad ke-19, yaitu Abbe Flandrin dan Abbe Michon, pada waktu itulah istilah “Grafologi” mulai diperkenalkan dan digunakan untuk menyebut ilmu yang mempelajari kepribadian berdasarkan tulisan tangan, dari hasil penelitiannya, ditemukan banyak korelasi dengan karakteristik kepribadian penulisnya.

Penelitian Abbe Michon mengenai analisis tulisan tangan dipublikasikan pada tahun 1872. Berawal dari waktu tersebut dan selanjutnya, minat di bidang Grafologi menyebar ke seluruh Eropa. Jadi, grafologi berkembang sebelum kelahiran psikologi.

Penelitian grafologi terus-menerus dilakukan oleh para penyelidik secara individual dan program-program di bawah arahan bantuan organisasi grafologi profesional seperti American Handwriting Analysis Foundation.

2.3 Manfaat Grafologi

Berdasarkan dari sejarahnya, ilmu grafologi digunakan untuk mengenali karakter seseorang dari tulisan tangannya. Saat ini penggunaan grafologi di Indonesia lebih umum digunakan di kalangan praktisi SDM (*Human Resource*), berikut adalah beberapa contoh penerapan grafologi di kalangan praktisi SDM [4].

- 1) Untuk keperluan rekrutmen.
- 2) Tes kepribadian.
- 3) *Assessment test*.
- 4) Tes untuk promosi.

Tidak hanya itu grafologi dapat digunakan lebih luas, contohnya:

- 1) Untuk pengenalan bakat dan minat.
- 2) Keperluan konseling, konsultasi karier
- 3) Kecocokan antarpasangan.
- 4) Analisis tindak kriminal.
- 5) Analisis penggunaan narkoba atau alkohol, dan lain-lain.

2.4 Fitur Tulisan Tangan Dalam Grafologi

Setiap tulisan yang ditorehkan mempunyai makna dan menunjukkan karakter penulis yang berbeda-beda. Hal ini dapat digunakan sebagai media untuk menganalisis kondisi kejiwaan atau kepribadian seseorang. Banyak sekali bentuk tulisan tangan serta masing-masingnya ciri khasnya sendiri. Namun dari sekian banyak tulisan dapat ditelusuri fitur dan ciri yang dapat mendeskripsikan sifat dan kepribadian tertentu [1].

Analisis tulisan tangan mempunyai kurang lebih dua puluh fitur untuk ditinjau yang dapat menggambarkan kemunculan kepribadian.

2.4.1 Tekanan Tulisan: Energi dan kesehatan

Penekanan tulisan dapat terjadi pada dua peristiwa, yaitu saat alat tulis pertama kali menyentuh kertas yang disebut tekanan primer, dan setelah tekanan primer dilakukan oleh si penulis sepanjang ia melanjutkan tulisannya. Dengan kata lain tekanan yang dimaksud adalah saat alat tulis pertama kali mulai menggoreskan tinta pada kertas. Tekanan tulisan ini mengindikasikan resistansi penulisnya [1].

Tabel 2.1 Kepribadian berdasarkan tekanan tulisan

Fitur	Ciri	Kepribadian
Tekanan tulisan	Tekanan ringan	Kurang memiliki keinginan untuk mencapai tujuan namun memiliki pemikiran yang tajam, mampu menghasilkan ide-ide.
	Tekanan sedang	Penulis memiliki kontrol yang baik terhadap emosi dan memiliki keseimbangan yang baik antara fisik dan mental.
	Tekanan kuat	Penulis menunjukkan keteguhan dan kegigihan dalam hidupnya, terdapat ketekunan, gairah, sifat agresif pada dirinya.

2.4.2 Margin tulisan: Masa Lalu, Sekarang, dan Masa Depan

Saat menulis dalam sehelai kertas, tentu semua orang akan berupaya agar tulisan tampak rapih dari berbagai tepi kertas, seperti dari tepi atas, tepi bawah, tepi kiri dan tepi kanan. Jika menggunakan komputer, merapikan tepi dapat dengan mudah diatur oleh penggunanya, tetapi tidak semudah itu jika menulis langsung. Tepi tulisan atau Margin antara tulisan individu satu dengan yang lainnya berbeda-

beda, semua pola yang di munculkan oleh si penulis dalam membuat garis semu (margin) memiliki arti tertentu, jika dilihat dari sudut grafologi [1].

2.4.2.1 Margin Kiri

Tepi kiri atau margin sebelah kiri menggambarkan bagaimana seseorang akan memulai sesuatu. Sebenarnya menulis dari tepi mana saja tidak dipermasalahkan, sebab hal ini merupakan pilihan penulis untuk memulainya dari mana. Tetapi ketidaksadaran diri dalam memilih titik awal penulisan turut mempengaruhi pilihan tersebut. Margin kiri terbagi atas kategori:

Tabel 2.2 Kepribadian berdasarkan margin kiri [4]

Fitur	Ciri	Kepribadian
Margin kiri	Margin kiri sempit	Latar belakang keluarga yang terbatas, rasa ekonomi dan tidakformalan.
	Margin kiri normal	Penulis dapat menyikapi masa lalunya dengan wajar, jadi bahan evaluasi, dan menjadi penyemangat untuk hidup yang sekarang.
	Margin kiri lebar	Memiliki kesadaran akan nilai-nilainya sendiri yang jadi suatu kebanggan.
	Margin kiri menyempit	Penulis punya kemauan dan usaha untuk meninggalkan masa lalunya, upaya bergerak maju untuk masa depan lebih baik, tetapi terkadang masih teringat akan masa lalu.
	Margin kiri melebar	Penulis ini berusaha meninggalkan masa lalunya terutama pengalaman-pengalaman buruk agar tidak memengaruhi dirinya sekarang dan nanti.
	Margin kiri tidak teratur	Penulis ingin melawan norma, tidak mau mengikuti aturan, tidak stabil, senang dengan permusuhan, tingkat kedisiplinan yang buruk.

2.4.2.2 Margin Kanan

Tepi kanan atau margin kanan menggambarkan sejauh mana penulis menggerakkan penanya dalam membuat tulisan, sampai akhirnya berhenti pada titik tertentu di bagian kanan kertas. Margin kanan mengungkap masa depan penulis. Margin kanan terbagi atas kategori [4]:

Tabel 2.3 Kepribadian berdasarkan margin kanan

Fitur	Ciri	Kepribadian
Margin Kanan	Margin kanan sempit	Mampu memanfaatkan ruang yang tersisa. Keberanian menghadapi kehidupan dan masalahnya, dan sanggup komunikatif.
	Margin kanan lebar	Perasaan cemas yang berlebihan, pesimistis terhadap masa depan, cenderung sering patah semangat.
	Margin kanan membentur	Penulis sangat bersemangat, menuruti kata hatinya, tidak terkontrol untuk mendapatkan keinginannya, tidak pernah belajar dari kesalahan, tidak peduli sekitar, suka menunda atau menolak yang akhirnya merugikan diri sendiri

2.4.2.3 Margin Atas

Seperti halnya margin kanan dan margin kiri, margin atas atau tepi kiri juga memiliki arti tersendiri. Margin atas adalah jarak atau tepi bagian atas kertas dengan baris pertama tulisan tangan dibuat. Margin atas mengindikasikan bagaimana penulis memosisikan diri dengan orang lain, margin atas terdiri dari atas kategori [1]:

Tabel 2.4 Kepribadian berdasarkan margin atas [4]

Fitur	Ciri	Kepribadian
Margin atas	Margin atas sempit	Penulis memosisikan dirinya di atas orang lain, mempunyai sifat kurang respek terhadap orang lain, egois, arogan, belum dewasa secara emosional.
	Margin atas normal	Penulis memosisikan dirinya setara dengan orang lain, tidak merendahkan dirinya didepan orang lain, juga tidak merendahkan orang lain.
	Margin atas lebar	Penulis memosisikan orang lain di atas dirinya, orang yang rendah hati, sikap respek yang tinggi, mampu menjalankan formalitas, dan mampu mengikuti kebiasaan umum.

2.4.2.4 Margin Bawah

Sama seperti margin atas atau tepi bagian atas, margin bawah juga dapat memberi petunjuk tentang diri penulisnya. Margin bawah adalah jarak atau tepi bagian bawah kertas dengan baris terakhir tulisan tangan dibuat. Margin bawah terdiri dari atas dua kategori [1]:

Tabel 2.5 Kepribadian berdasarkan margin bawah

Fitur	Ciri	Kepribadian
Margin bawah	Margin bawah sempit	Mengindikasikan orang yang komunikatif, realistis, kelompok yang sentimental dan pemimpi.
	Margin bawah lebar	Mengindikasikan orang yang bersikap pura-pura, tidak serius, tidak peduli, idealisme, dan kepekaan estetik.

2.4.3 Spasi: Interaksi dengan Lingkungan

Di dalam tulisan, pemisah atau jarak antar kata disebut dengan spasi. Jarak atau spasi dalam tulisan dapat dibedakan menjadi tiga, yaitu jarak antar kata dan jarak antar baris. Ketiga macam spasi atau jarak tersebut memiliki penafsiran yang berbeda [1].

2.4.3.1 Spasi Antarkata

Spasi antar kata adalah jarak antara satu kata dengan kata berikutnya. Spasi antarkata merupakan cerminan, bagaimana pikiran mengalir begitu saja (alami), seperti sedang berbicara ada jeda dalam pembicaraan. Spasi antarkata juga dapat mengungkap interaksi penulis dengan lingkungannya (pergaulan maupun sosial). Spasi antar kata terdiri atas kategori [1].

Tabel 2.6 Kepribadian berdasarkan spasi antar kata

Fitur	Ciri	Kepribadian
Spasi antar kata	Spasi antar kata sempit	Penulis dapat cepat menjalin relasi dengan orang lain, juga mudah akrab dengan orang lain, pandai bergaul dan disukai di lingkungan sosialnya..
	Spasi antar kata medium	Penulis berada di lingkungan sosial sesuai kebutuhan atau menyendiri bila diperlukan, dapat

Fitur	Ciri	Kepribadian
		menyesuaikan diri dengan kedua kondisi tersebut.
	Spasi antar kata lebar	Penulis tidak menyukai kegiatan sosial, tidak suka keramaian, tidak merasa nyaman dengan orang banyak, dan mengikuti kegiatan sosial dengan keterpaksaan.

2.4.3.2 Spasi Antarhuruf

Spasi antarhuruf adalah jarak antara satu huruf dengan huruf berikutnya. Spasi antarkata mengungkap kebutuhan interaksi atau keakraban secara fisik pada penulisnya. Spasi antarhuruf terdiri atas kategori [4].

Tabel 2.7 Kepribadian berdasarkan spasi antar huruf

Fitur	Ciri	Kepribadian
Spasi antar huruf	Spasi antar baris huruf medium	Penulis dapat menerima kontak fisik secara akrab, namun dengan hanya dengan orang-orang yang membuatnya nyaman.
	Spasi antar huruf sempit	Penulis senang akrab secara fisik, merangkul atau sebaliknya.
	Spasi antar huruf lebar	Penulis tidak suka kontak atau akrab secara fisik, menginginkan ada ruang antara dirinya dan orang lain.

2.4.3.3 Spasi Antarbaris

Spasi antarbaris adalah ruang diantara baris tulisan di atas dengan baris tulisan dibawahnya. Spasi antarbaris ini menggambarkan kemampuan pengorganisasian, pengaturan, dan kerapian penulisnya. Spasi antarbaris terbagi atas kategori [4]:

Tabel 2.8 Kepribadian berdasarkan spasi antar baris

Fitur	Ciri	Kepribadian
Spasi antar baris	Spasi antar baris teratur	Penulis merupakan perencana yang baik (<i>well planer</i>), dan orang yang rapi.
	Spasi antar baris tidak teratur	Penulis mempunyai sifat kurang rapih, dan bukan perencana yang baik.

2.4.4 Ukuran Huruf: Harga Diri

Secara garis besar, ukuran tulisan dalam grafologi dibedakan menjadi tiga, yaitu kecil, sedang, dan besar. Besar kecilnya huruf pada tulisan memberikan gambaran bagaimana seseorang secara tidak sadar melihat dirinya sendiri, apakah termasuk orang yang percaya diri atau kurang yakin pada dirinya. Selain itu tulisan tangan sering kali digunakan untuk mengindikasikan tingkat intelektual penulisnya [1].

Tabel 2.9 Kepribadian berdasarkan ukuran huruf [4]

Fitur	Ciri	Kepribadian
Ukuran huruf	Ukuran huruf besar	Cocok bekerja dilapangan, atau pekerjaan dengan mobilitas tinggi. Penulis juga membutuhkan pengakuan dari orang lain, tidak mampu berkonsentrasi untuk waktu yang panjang, mempunyai jiwa petualang dan butuh kebebasan, mempunyai sikap ekspresif, optimis, dan egois.
	Ukuran huruf sedang	penulis ini cocok bekerja di kantor, administrasi, pekerjaan dengan ketelitian dan rutinitas. Penulis juga mampu berkonsentrasi dalam waktu yang panjang, mampu bekerja secara rutin, ketat dalam pengeluaran uang, mempunyai sikap pemalu, realistis, dan teliti.
	Ukuran huruf kecil	penulis cocok bekerja dengan kombinasi belakang meja serta lapangan. Penulis ini memiliki harga diri, karakter, dan kepribadian yang baik.

2.4.5 Arah Kata (Kemiringan): Ekspresi Diri

Pola ayunan kata menjadi arah kemana penulis mengekspresikan dirinya, apakah ke dalam atau ke luar diri penulis. Arah huruf juga mengekspresikan perasaan dan emosional penulisnya [4].

Tabel 2.10 Kepribadian berdasarkan arah kata (kemiringan)

Fitur	Ciri	Kepribadian
Kemiringan	Kemiringan ke kanan	penulis senang bergabung dalam kelompok sosial, senang menjalin hubungan dengan orang lain, menganggap hubungan sosial itu penting. Penulis mudah mengekspresikan dirinya, ekstrovert, mudah berempati.
	Kemiringan tegak	penulis merupakan orang yang rasional, mengutamakan logika daripada perasaan, mampu mengontrol perasaan, mampu mengambil keputusan dengan rasional dan mengabaikan perasaan.
	Kemiringan ke kiri	penulis membutuhkan kenyamanan dan ruang bagi dirinya, menjaga perasaan dan pikirannya dari orang lain. Penulis merupakan introvert dan berorientasi pada kebutuhan diri.

2.4.6 Garis Dasar: Tingkat Motivasi

Garis dasar atau dapat disebut *baseline* adalah garis khayal atau imajiner yang terbentuk ketika menarik garis paling kiri menuju paling kanan. Garis dasar adalah pergerakan motivasi, termasuk ada emosional di dalamnya dari masa lalu, sekarang, menuju masa depan. Dalam hal ini garis yang terbentuk menunjukkan tingkat motivasi penulis dalam perjalanan menuju masa depan atau mencapai tujuan hidupnya [4].

Tabel 2.11 Kepribadian berdasarkan garis dasar

Fitur	Ciri	Kepribadian
Garis dasar	Garis dasar lurus	Penulis merupakan seseorang yang stabil dalam berperilaku, konsisten, dan terkendali. Seringkali menunjukkan ketenangan, penulis juga memiliki kontrol yang baik terhadap dirinya, termasuk kontrol suasana hatinya.
	Garis dasar menaik	Mengungkapkan semangat yang ceria, ambisi, optimisme, kejujuran,

Fitur	Ciri	Kepribadian
		dan motivasi untuk sukses. Penulis mudah bergairah dan mudah mengambil tindakan.
	Garis dasar menurun	Penulis merupakan seseorang yang lelah, depresi, sedih, atau pesimisme. Seringkali menunjukkan penurunan semangat dan daya juang, ada kekecewaan, keputusan, atau ketidakbahagiaan.

2.4.7 Zona Tulisan Tangan: Kepribadian, Waktu, dan Tubuh.

Pada dasarnya pembagian zona mengacu pada prinsip *antropomorfis*, yaitu prinsip pembagian tubuh manusia menjadi tiga bagian, diantaranya kepala (atas), badan (tengah), dan kaki (bawah), ketiga bagian tersebut dibagi sama besar [4].

Tabel 2.12 Kepribadian berdasarkan zona tulisan

Fitur	Ciri	Kepribadian
Zona tulisan	Dominan di zona atas	Penulis memerhatikan pemikiran, gagasan, dan kehidupan spritual. Penulis lebih mengandalkan fikiran daripada fisik, suka berdiskusi tentang teori, konsep, dan gagasan.
	Dominan di zona tengah	Penulis lebih memerhatikan kehidupan, kondisi, penampilan diri saat ini, juga memiliki sosial yang aktif.
	Dominan di zona bawah	Penulis lebih memperhatikan kebutuhan dasar kehidupan seperti makanan, materi, dan seksualitas.

2.4.8 Huruf Istimewa: Ciri-Ciri Penting Lainnya

2.4.8.1 Huruf Istimewa: Huruf ‘i’

Tabel 2.13 Kepribadian berdasarkan huruf istimewa: huruf ‘i’

Fitur	Ciri	Kepribadian
Huruf istimewa huruf ‘i’	Titik bundar dan tipis	Setia, dapat dipercaya, dan mempunyai kesetiakawanan.
	Titik yang panjang dan miring	Adanya gangguan dari orang lain, jahil atau iseng.
	Titik ke bawah	Orang yang tidak suka pada dirinya sendiri.

Fitur	Ciri	Kepribadian
	Titik miring ke kiri	Indikasi gangguan terhadap sesuatu dimasa lalu.
	Titik di kanan	Orang yang tidak sabar, terburu-buru.
	Titik tepat di atas huruf	Daya ingat sangat baik, detail dan cermat.
	Titik di kiri	Orang yang suka menunda pekerjaan.
	Titik berbentuk lingkaran	Ingin mendapatkan perhatian.

2.4.8.2 Huruf istimewa: Huruf 'o'

Tabel 2.14 Kepribadian berdasarkan huruf istimewa: huruf 'o'

Fitur	Ciri	Kepribadian
Huruf istimewa huruf 'o'	Loop lingkaran di atas	Suka merahasiakan sesuatu.
	Loop lingkaran di kanan	Kebohongan pada oranglain.
	Loop lingkaran di kiri	Kebohongan pada orang diri sendiri.
	Tanpa loop lingkaran	Orang yang jujur, hati-hati sebelum bicara, mampu menjaga rahasia.
	Double loop	Suka berbicara, bawel.
	Ada bagian terbuka	Kebiasaan berbohong.

2.4.8.3 Huruf Istimewa: Huruf 't'

Tabel 2.15 Kepribadian berdasarkan huruf istimewa: huruf 't'

Fitur	Ciri	Kepribadian
Huruf istimewa huruf 't'	Stem t rendah	Kepercayaan diri dan target pribadi rendah.
	Stem t di tengah	Kepercayaan diri dan target pribadi cukup baik.
	Stem t tinggi	Kepercayaan diri dan target pribadi tinggi.
	Stem t di puncak huruf	Kepercayaan diri dan target pribadi sangat tinggi, terkesan sombong.
	Stem t dengan loop	Sensitif terhadap perlakuan orang lain.
	t bentuk tenda	Keras kepala.
	Stem t bentuk payung	Luwes dan dinamis.
	t bersambung ke huruf berikutnya	Mawas diri, kesadaran diri baik.

2.4.8.4 Huruf Istimewa: Huruf ‘g’, ‘y’, dan ‘j’

Tabel 2.16 Kepribadian berdasarkan huruf istimewa: huruf ‘g’, ‘y’, dan ‘j’

Fitur	Ciri	Kepribadian
Coretan akhir	Coretan akhir mengarah ke bawah	Ketakutan untuk menggapai kesuksesan.
	Coretan akhir mengarah ke atas	Dorongan untuk mencapai target dan keuangan yang besar.
	Tanpa loop	Terkadang takut dan terkadang semangat dalam menggapai kesuksesan.

2.4.9 Awalan dan Akhiran: Kemandirian

Dalam menulis ada orang-orang yang membuat coretan awal (awalan) di depan huruf, dan tentu ada yang tidak. Ada orang-orang yang membuat coretan akhir (akhir) di akhir huruf, dan tentu ada yang tidak.

2.4.9.1 Coretan awal

Tabel 2.17 Kepribadian berdasarkan coretan awal

Fitur	Ciri	Kepribadian
Coretan awal	Coretan awal yang panjang	Ragu-ragu, kerap menutupi kesalahan yang diperbuat, membiarkan orang lain berada di depannya, dan tidak suka perubahan.
	Coretan awal pada huruf kapital	Tergolong orang yang suka pamer kemampuan yang dimilikinya, dan sangat memperhatikan penampilan fisik.
	Coretan awal pada zona atas	Memiliki imajinasi yang sepenuhnya tidak dapat diwujudkan, tetapi tergolong orang yang mampu mempersiapkan diri dengan baik sebelum menunjukkan kemampuannya.
	Coretan awal pada zona tengah	Ambisius dengan energi yang berlebihan, sangat mempertahankan pendapat.
	Tanpa coretan awal	Mandiri, dewasa, kreatif, mempunyai pendekatan yang baik dalam menyelesaikan persoalan.

2.4.9.2 Coretan akhir

Tabel 2.18 Kepribadian berdasarkan coretan akhir

Fitur	Ciri	Kepribadian
Coretan akhir	Coretan akhir yang panjang	Indikasi rasa antusias, dorongan semangat yang kuat.
	Coretan akhir menabrak kata berikutnya	Lebih memikirkan diri sendiri daripada orang lain serta tidak mudah percaya pada orang lain.
	Coretan akhir yang tidak selesai	Mempunyai kebiasaan meninggalkan sesuatu yang diinginkan atau dikerjakan dan kurang menghargai kewajiban dalam pergaulan sosial.
	Coretan akhir yang mengayun	Egois, matrealistis, dan tampil begitu percaya diri padahal aslinya tidak begitu.
	Coretan akhir dengan aksesoris	Keras kepala, tidak patuh, dan nampak konvensional padahal aslinya tidak begitu.
	Coretan akhir jatuh ke bawah	Keras terhadap apa yang diinginkan dan tidak diinginkan, murah hati dan royal.
	Tanpa coretan akhir	Mandiri, tidak mencari hubungan sosial, tidak mementingkan penerimaan orang lain, dan berhati-hati dalam tindakan dan perkataan.

2.4.10 Ketersambungan huruf: Sistem Berfikir

Ketersambungan atau biasa disebut *connection* adalah cara penulis membuat rangkaian antara satu huruf dengan huruf berikutnya. Berikut adalah kepribadian penulis jika dilihat berdasarkan ketersambungan huruf.

Tabel 2.19 Kepribadian berdasarkan ketersambungan huruf

Fitur	Ciri	Kepribadian
Ketersambungan huruf	Huruf tersambung	Penulis biasanya menjalankan sistem <i>first in first out</i> , memiliki logika berpikir yang baik, cerdas, mau bekerja sama, dan mampu beradaptasi.
	Huruf terpisah	Penulis biasanya mempunyai konsep <i>multitasking</i> , dapat menyelesaikan beberapa hal secara acak atau bersamaan, observasi yang baik, mandiri, dan berhati-hati.

Fitur	Ciri	Kepribadian
	Kombinasi tersambung dan terpisah	Penulis biasanya berfikir dengan skala prioritas, mampu memilah pekerjaan yang mana harus dikerjakan terlebih dahulu.

2.4.11 Huruf Kapital: Kepercayaan Diri

Huruf kapital dapat dibilang kerabat dekat dari ukuran huruf. Bila ukuran huruf mengungkap harga diri penulisnya, secara umum huruf kapital secara khusus mengungkap kepercayaan diri penulisnya.

Tabel 2.20 Kepribadian berdasarkan huruf kapital

Fitur	Ciri	Kepribadian
Huruf kapital	Kapital dengan ukuran besar	Pribadi yang percaya diri, berusaha mencari status, merasa dirinya pantas, dan menginginkan kekuasaan.
	Kapital salah tempat	Seseorang yang emosional, emosi tidak stabil, mementingkan pemikiran, tindakan, serta ide-idenya.
	Penggunaan kapital berlebihan	Kerap bingung dalam penentuan konsep, tentang apa yang harus di prioritaskan, dan eksentris.
	Kapital berukuran kecil	Seseorang yang mengabaikan diri sendiri, sederhana, pemalu, dan peduli pada orang lain.
	Kapital tinggi dan sempit	Sempit, emosional, pemalu, ambisius tapi tertahan karena menarik diri
	Sempit namun normal dengan ukuran huruf lain	Kerap kecewa, mudah ccuriga, dan sangat berhati-hati.
	Kapital dengan ukuran lebar	Percaya pada diri sendiri, imajinatif, dan ramah.

2.5 Pengertian Citra

Sebuah piksel adalah sampel dari pemandangan yang mengandung intensitas citra yang dinyatakan dalam bilangan bulat. Citra sendiri dapat didefinisikan sebagai fungsi dari dua variabel misalnya ax,y di mana a merupakan amplitudo atau kecerahan, sedangkan x, y adalah koordinat yang membentuk citra [6].

Dalam istilah lainnya, citra juga dapat diartikan sebagai gambar pada salah satu komponen multimedia dengan karakteristik yang tidak dimiliki oleh data berbentuk teks. Secara harfiah citra adalah gambar atau sebuah terusan cahaya pada bidang dwimatra (dua dimensi) yang ditinjau dari sudut pandang matematis. [6]

Secara umum, ada tiga jenis citra yang digunakan dalam pemrosesan citra. Ketiga jenis citra tersebut adalah citra berwarna (RGB), citra berskala keabuan (*grayscale*), dan citra biner [7].

1) Citra berwarna

Citra berwarna, atau disebut juga sebagai citra RGB, merupakan jenis citra yang menyajikan warna dalam bentuk komponen yang meliputi *Red* (merah), *Green* (hijau), dan *Blue* (biru). Setiap komponen warna menggunakan nilai delapan bit yang berarti nilainya berkisar antara 0 sampai 255. Oleh karena itu, kemungkinan warna yang dapat dihasilkan oleh mencapai $255 \times 255 \times 255$ atau sebanyak 16.581.375 warna.

2) Citra berskala keabuan

Sesuai dengan namanya, citra berskala keabuan hanya menangani gradasi warna hitam dan putih, yang menghasilkan efek warna abu-abu. Pada citra berskala keabuan, warna dinyatakan dengan intensitas. Dalam hal ini, intensitas berkisar antara 0 sampai dengan 255. Nilai 0 menyatakan hitam, sedangkan 255 menyatakan putih.

3) Citra biner

Citra biner adalah citra dengan setiap piksel hanya dinyatakan dengan sebuah nilai dari dua kemungkinan, yaitu antara nilai 0 dan 1. Nilai 0 menyatakan warna hitam dan nilai 1 menyatakan warna putih. Citra jenis ini banyak dipakai untuk pemrosesan citra, misalnya untuk kepentingan memperoleh tepi bentuk suatu objek.

2.6 Pengolahan Citra Digital

Secara umum, menurut Jain, istilah pengolahan citra digital menyatakan “pemrosesan gambar berdimensi-dua melalui komputer digital. Sedangkan menurut Efford pengolahan citra adalah istilah umum untuk berbagai teknik yang keberadaannya untuk memanipulasi dan memodifikasi citra dengan berbagai cara.

Foto adalah contoh gambar berdimensi dua yang dapat diolah dengan mudah. Setuap foto dalam bentuk citra digital dapat diolah melalui perangkat lunak tertentu [7].

Dalam pengolahan citra digital terlebih untuk citra yang akan digunakan sebagai data masukan suatu proses, perlu memiliki kualitas yang baik. Contoh hal-hal yang mengganggu kualitas citra diantaranya adanya bercak hitam dan putih yang bukan bagian dari objek pada citra.

Meskipun demikian, untuk dapat memperoleh citra yang memiliki kualitas yang baik memerlukan alat optik yang bagus dengan harga yang relatif mahal. Alternatif dari penggunaan alat optik yang mahal adalah dengan dilakukannya pengolahan citra dengan berbagai macam metode, salah satunya adalah menggunakan konsep *smoothing*. Pengolahan gambar atau pengolahan citra yang sering disebut *image processing* adalah suatu proses untuk mengolah gambar yang mengubah gambar menjadi gambar lain yang memiliki kualitas lebih baik untuk tujuan tertentu.

Tujuan utama dari pengolahan citra adalah [8]:

- 1) Memperbaiki kualitas citra, dimana citra yang dihasilkan dapat menampilkan informasi dengan lebih 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.

2.6.1 Grayscale

Citra *grayscale* seringkali disebut sebagai citra keabuan. Citra *grayscale* menangani gradasi warna hitam dan putih yang menghasilkan efek warna abu-abu. Gradasi warna pada setiap gambar dinyatakan sebagai intensitas. Setelah citra berwarna diubah kedalam bentuk *grayscale*, intensitas yang terbentuk berkisar diantara 0 sampai dengan 255. Nilai 0 menyatakan hitam dan nilai 255 menyatakan putih. Pada penelitian ini perubahan citra berwarna kedalam bentuk *grayscale* dilakukan dengan menggunakan persamaan 2.1 [7].

$$\text{Grayscale} = 0,2989 * R + 0,5870 * G + 0,1141 * B \quad (2.1)$$

Dimana nilai R menyatakan nilai piksel merah, G menyatakan piksel hijau, dan B menyatakan nilai komponen biru. Contoh hasil dari teknik *grayscale* sendiri telah tergambarkan pada Gambar 2.1.



Gambar 2.1 Contoh hasil teknik *grayscale*

2.6.2 *Threshold*

Threshold atau biasa disebut sebagai pengambangan citra merupakan teknik untuk menghasilkan citra biner, yaitu citra yang hanya mempunyai dua tingkat keabuan (hitam dan putih saja). Saat ini terdapat dua teknik pengambangan, yaitu pengambangan global (*global treshold*) dan pengambangan secara lokal adaptif (*local adaptive treshold*), pada penelitian ini digunakan sebuah titik tengah atau pengambangan global. Dimana pengambangan global dapat dilakukan dengan persamaan berikut [9]:

$$\text{treshold}_{x,y} = \begin{cases} 255 & \text{jika } a_{x,y} \geq 127 \\ 0 & \text{jika tidak} \end{cases} \quad (2.2)$$

Contoh dari gambar *threshold* sendiri telah tergambarkan pada Gambar 2.2.



Gambar 2.2 Contoh hasil teknik *threshold*

2.6.3 Segmentasi

Segmentasi adalah sebuah teknik untuk mengelompokkan citra menjadi beberapa wilayah (region) dengan kriteria tertentu, atau dengan kata lain merupakan proses pembagian citra ke dalam wilayah yang mempunyai kesamaan fitur berupa *grayscale, texture, color atau motion*.

Dari pengertiannya, segmentasi mempunyai tujuan untuk menemukan karakteristik khusus yang dimiliki oleh suatu citra. Oleh karenanya, segmentasi dibutuhkan untuk pengenalan pola dimana jika proses segmentasi baik maka kualitas dalam pengenalan polanya juga akan semakin baik. Pada penelitian ini akan menggunakan segmentasi baris [9].

Segmentasi baris dilakukan untuk mencari fitur baris pada tulisan, segmentasi baris pada penelitian ini dijadikan acuan untuk penentuan fitur yang lainnya. Segmentasi baris dilakukan dengan dengan pencarian batas atas baris dan batas bawah dengan cara segmentasi horizontal. Piksel putih yang ditemukan pertama akan dijadikan batas atas, dan piksel putih akan dijadikan batas bawah jika setelahnya hanya ditemukan baris berisi piksel hitam, isi karakter diantara batas awal dan akhir akan dinyatakan sebagai baris, proses ini terus diulang hingga tidak ditemukan lagi piksel putih.

2.6.4 Ekstraksi Fitur

Ekstraksi fitur (*feature extraction*) adalah proses mendapatkan nilai yang bersifat karakteristik, dan sebagai input dari pembelajaran mesin [23]. Ekstraksi fitur juga biasa disebut sebagai pencirian, yang merupakan proses mengubah tiap-tiap sampel sinyal menjadi vektor-vektor data [10].

Fitur yang baik sebisa mungkin harus memenuhi persyaratan berikut [9]:

- 1) Dapat membedakan suatu objek dengan yang lain (*discrimination*).
- 2) Memperhatikan kompleksitas komputasi dalam memperoleh fitur. Kompleksitas komputasi yang tinggi tentu akan menjadi beban tersendiri dalam menemukan suatu fitur.
- 3) Tidak terikat (*independence*) dalam arti bersifat invarian terhadap berbagai transformasi (rotasi, penskalaan, pergeseran, dan lain sebagainya).

4) Jumlahnya sedikit, fitur yang jumlahnya sedikit akan dapat menghemat waktu komputasi dan ruang penyimpanan untuk proses selanjutnya.

Metode yang akan digunakan untuk pencirian atau ekstraksi fitur dalam penelitian ini adalah dengan menggunakan metode *Zoning zone centroid zone*.

2.6.4.1 Ekstraksi Fitur Margin Kiri

Ekstraksi fitur margin kiri dilakukan untuk mencari fitur margin kiri tulisan. Segmentasi ini merupakan tahap lanjutan dari segmentasi baris. Proses segmentasi ini adalah:

- a) Scan secara vertikal dari kiri untuk menemukan piksel putih terdekat dari tepi kiri kertas pada tiap-tiap barisnya.
- b) Jika sudah ditemukan, hitung jarak piksel hitam dari tepi kiri kertas hingga ke piksel putih yang ditemukan.

2.6.4.2 Ekstraksi Fitur Margin Kanan

Proses ekstraksi fitur margin kanan hampir sama dengan segmentasi tepi kiri, segmentasi ini dilakukan untuk mencari fitur margin kanan tulisan. Proses segmentasi ini adalah:

- a) Scan secara vertikal dari kanan untuk menemukan piksel putih terdekat dari tepi kanan kertas pada tiap-tiap barisnya.
- b) Jika sudah ditemukan, hitung jarak piksel hitam dari tepi kanan kertas hingga ke piksel putih yang ditemukan.

2.6.4.3 Ekstraksi Fitur Spasi Antar Kata

Ekstraksi fitur spasi antar kata dilakukan untuk mencari fitur spasi antar kata, proses ekstraksi fitur spasi antar kata dilakukan hampir sama dengan segmentasi baris. Berikut adalah proses segmentasi kata:

- a) Scan secara vertikal pada baris untuk menemukan piksel putih yang akan dianggap sebagai batas atas.
- b) Jika ditemukan, selanjutnya scan kembali hingga tidak ditemukan piksel putih.

- c) Jika ditemukan, hitung jarak piksel hitam tersebut, jika jarak piksel hitam melewati 13 piksel maka dinyatakan sebagai spasi kata, jika tidak maka masih dinyatakan sebagai satu kata yang sama.
- d) Jika telah ditemukan spasi kata, piksel putih terakhir dinyatakan sebagai batas akhir kata.
- e) Proses dilakukan satu baris penuh hingga tidak ditemukan lagi piksel yang dianggap kata.

2.6.4.4 Ekstraksi Fitur Spasi Antar huruf

Ekstraksi fitur spasi antar huruf dilakukan untuk mencari fitur spasi antar kata, proses ekstraksi fitur spasi antar huruf dilakukan hampir sama dengan segmentasi baris. Berikut adalah proses ekstraksi fitur spasi antar huruf:

- a) Scan secara vertikal pada baris untuk menemukan piksel putih yang akan dianggap sebagai batas atas.
- b) Jika ditemukan, selanjutnya scan kembali hingga tidak ditemukan piksel putih.
- c) Jika ditemukan, hitung jarak piksel hitam tersebut, jika piksel hitam melewati batas maka dinyatakan sebagai spasi kata, jika tidak maka masih dinyatakan sebagai satu kata yang sama.
- d) Jika telah ditemukan spasihuruf, piksel putih terakhir dinyatakan sebagai batas akhir huruf.
- e) Proses dilakukan pada kata di satu baris penuh.

2.6.4.5 Ekstraksi Fitur Zoning

Ekstraksi fitur zoning digunakan untuk mencari nilai fitur kemiringan kata dan garis dasar. Metode *zoning* memiliki beberapa keunggulan jika dibandingkan dengan metode lainnya, diantaranya adalah penciriannya yang sederhana, kompleksitas yang rendah dan memiliki perhitungan yang cepat. Menurut Rajashekararadya dan Ranjan (2008) ada beberapa variasi algoritma untuk metode ekstraksi *zoning*, yaitu ICZ (*image centroid and zone*), ZCZ (*zone centroid zone*), dan algoritma gabungan ICZ dan ZCZ [10]. Pada penelitian ini metode yang digunakan menggunakan metode ZCZ, berikut adalah tahapan metode ZCZ [11]:

- 1) Dengan menggunakan algoritma *zone centroid zone* pertama-tama citra masukan akan dibagi menjadi n zona yang sama. Kemudian, hitung *zone centroid* dari masing-masing zona menggunakan persamaan 2.7 dan 2.8 berikut:

$$X_c = \frac{\sum_{i=1}^x f(x,y)x_i}{\sum f(x,y)} \quad (2.3)$$

$$y_c = \frac{\sum_{i=1}^y f(x,y)y_i}{\sum f(x,y)} \quad (2.4)$$

Dengan $f(x, y)$ adalah nilai piksel dari citra pada posisi tertentu.

- 2) Kemudian setelah dilakukan perhitungan keseluruhan piksel, hitung jarak antara *centroid* citra dengan setiap piksel yang ada dalam zona menggunakan persamaan 2.9 berikut:

$$d_{(P,C)} = \sqrt{(x_p - x_c)^2 + (y_p - y_c)^2} \quad (2.5)$$

Dengan d adalah jarak antara dua titik, P adalah koordinat titik berat, C adalah koordinat piksel, x_p = koordinat x titik berat, y_p = koordinat y titik berat, x_c = koordinat x piksel, dan y_c = koordinat y piksel.

- 3) Setelah keseluruhan perhitungan jarak antara *centroid* dengan setiap piksel dalam zona sudah dilakukan, berikutnya adalah mencari nilai fitur dengan cara merata-ratakan total jarak tersebut dengan persamaan:

$$Rata - rata = \frac{jarak\ 1 + jarak\ 2 + jarak\ 3 + \dots + jarak\ n}{Banyak\ titik} \quad (2.6)$$

2.7 Support Vector Machine

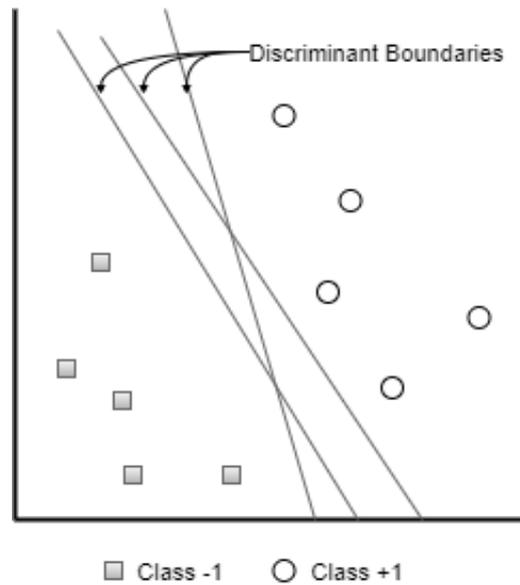
Support Vector Machine (SVM) pertama kali diperkenalkan oleh Vapnik pada tahun 1992 sebagai teknik klasifikasi yang efisien untuk masalah nonlinier. SVM berbeda dengan teknik klasifikasi pada era 1980-an, seperti *decision tree* dan ANN, yang secara konsepnya kurang begitu jelas dan seringkali terjebak pada optimum lokal. SVM berusaha menemukan *hyperplane* dengan memaksimalkan jarak antar kelas. Dengan cara ini, SVM dapat menjamin kemampuan generalisasi yang tinggi untuk data-data yang akan datang [12].

Pada perkembangannya SVM dapat diperluas untuk klasifikasi multi kelas dengan sejumlah skema. *Binary Class SVM*, suatu model dasar SVM untuk menyelesaikan masalah klasifikasi dua kelas. Berikutnya *Multi Class SVM*, model yang mampu menangani klasifikasi lebih dari dua kelas [13]. Dan *Fuzzy SVM*, yang secara matematis lebih baik dalam menangani klasifikasi lebih dari dua kelas [14].

Pada SVM ini, *hyperplane* terbaik adalah *hyperplane* yang mempunyai margin maksimal yang diperoleh dari alternatif garis pemisah (*discriminant boundaries*). *Margin* adalah jarak antara *hyperplane* dengan titik terdekat dari masing-masing kelas, sedangkan *support vector* sendiri merupakan titik terdekat tersebut [15].

Berbeda dengan *neural network* yang tujuannya adalah mencari *hyperplane* pemisah antar kelas, pada dasarnya SVM bertujuan untuk menemukan *hyperplane* terbaik pada *input space*. SVM mempunyai prinsip dasar *linier classifier*, yang selanjutnya dikembangkan agar dapat bekerja pada permasalahan non-linier dengan memasukkan konsep *kernel trick* pada ruang kerja berdimensi tinggi. Perkembangan ini memberikan rangsangan minat penelitian di bidang *pattern recognition* untuk menggali lebih dalam potensi dan kemampuan SVM secara teoritis maupun dari segi aplikasi. Pada saat ini SVM telah berhasil diaplikasikan dalam problema dunia nyata (*real world problem*), dan secara umum memberikan solusi yang lebih baik jika dibandingkan dengan metode konvensional lainnya, misalnya *artificial neural network*.

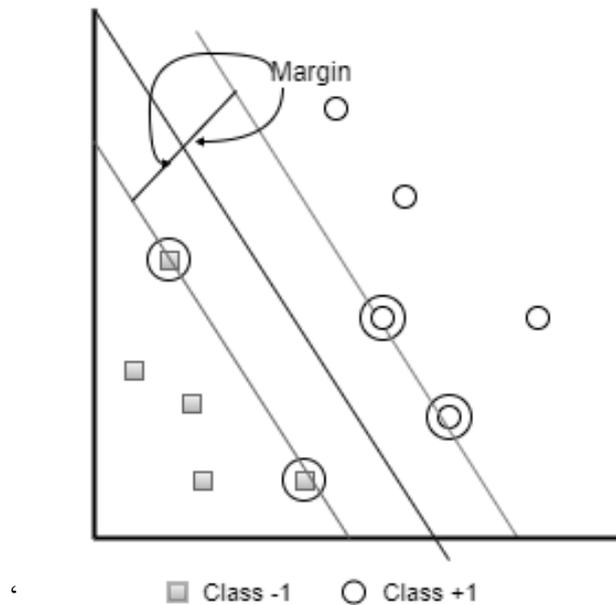
Konsep SVM akan dijelaskan secara sederhana sebagai usaha mencari *hyperplane-hyperplane* terbaik dengan fungsi sebagai pemisah dua buah *class* pada *input space*. Gambar 2.3 memperlihatkan *pattern* yang merupakan anggota dari dua buah kelas, dengan kelas +1 dan -1 [16]:



Gambar 2.3 SVM berusaha menemukan *hyperplane* terbaik

Pattern yang tergabung pada *class* -1 disimbolkan dengan warna abu (kotak), sedangkan *pattern* +1 disimbolkan dengan warna putih (lingkaran). Permasalahan klasifikasi dapat diterjemahkan dengan cara menemukan *hyperplane* (garis) yang memisahkan antara kedua kelas tersebut. Contoh berbagai alternatif (*Discriminant Boundaries*) ditunjukkan pada gambar 2.3.

Hyperplane pemisah terbaik antara kedua kelas dapat ditemukan dengan cara mengukur margin *hyperplane* tersebut, kemudian mencari titik maksimalnya. *Pattern* terdekat (*support vector*) ditunjukkan pada gambar 2.4, dimana garis yang berwarna solid menunjukkan *hyperplane* terbaik [16].



Gambar 2.4 SVM mencari *hyperplane* terbaik

Pada ruang berdimensi tinggi, konsep SVM adalah mencari *hyperplane* yang dapat memaksimalkan jarak (margin) antara kelas data. Persamaan garis lurus dapat dituliskan sebagai berikut:

$$y - ax - b = w * x - b \quad (2.7)$$

Data latih dinyatakan oleh x_i, y_i , dimana x_i ($x_i = \{x_1, x_2, \dots, x_n\}$) merupakan atribut (fitur) set untuk data latih. Untuk y_i ($\{-1, 1\}$) menyatakan label kelas. Definisi persamaan suatu *hyperplane* pemisah dapat dituliskan dengan persamaan 2.8 berikut:

$$w^T * x_i + b = 0 \quad (2.8)$$

Sebuah data x_i diklasifikasikan sebagai kelas 1 jika:

$$w^T * x_i + b \leq 0 \text{ untuk } y_i = +1 \quad (2.9)$$

Dan diklasifikasikan sebagai -1 jika:

$$w^T * x_i + b \geq 0 \text{ untuk } y_i = -1 \quad (2.10)$$

Dimana :

x_i = Data input w = Nilai dari bidang normal
 y_i = label b = posisi bidang relatif terhadap kordinat

Parameter w merupakan parameter yang akan dicari nilainya. Jika label data $y_i = 1$, maka pembatasnya akan seperti persamaan 2.11 berikut:

$$w^T * x_i + b \leq 1 \quad (2.11)$$

Jika label data $y_i = -1$, maka pembatas akan seperti persamaan 2.12 berikut:

$$w^T * x_i + b \leq -1 \quad (2.12)$$

Untuk mendapatkan *hyperlane* terbaik adalah dengan mencari *hyperplane* yang berada di tengah-tengah antara dua bidang pembatas kelas dan untuk mendapatkannya, bisa dengan memaksimalkan margin atau jarak antara dua set objek dari kelas yang berbeda. Margin dapat dihitung dengan $\frac{2}{\|w\|}$

Mencari *hyperplane* terbaik dapat menggunakan metode *Quadratic Programming (QP) Problem* yaitu meminimalkan:

$$\min r(w) = \frac{1}{2} w^T w \quad (2.13)$$

Dengan batasannya adalah:

$$y_i(w^T * x_i + b) - 1 \geq 0, \quad i = 1, 2, 3, \dots, n \quad (2.14)$$

Solusi untuk mengoptimasi oleh Vapnik (1995) diselesaikan menggunakan metode *lagrange* sebagai berikut:

$$L(w, b, a) = \frac{1}{2} w^T w - \sum_{i=1}^n a_i \{y_i [(w^T * x_i) + b] - 1\} \quad (2.15)$$

Dengan a_i = pengganda fungsi *lagrange* dan $i = 1, 2, \dots, n$

Nilai optimal dapat dihitung dengan memaksimalkan L terhadap a_i , dan meminimalkan w dan b . Hal ini layaknya kasus dual problem.

$$\frac{\partial L}{\partial b} = w - \sum_{i=1}^n a_i y_i = 0 \quad (2.16)$$

$$\frac{\partial L}{\partial w} = \sum_{i=1}^n a_i x_i y_i = 0 \quad (2.17)$$

$$\frac{\partial L}{\partial w} = w - \sum_{i=1}^n a_i y_i (w^T x_i + b) - \sum_{i=1}^n a_i = 0 \quad (2.18)$$

Maka masalah *lagrange* untuk klasifikasi dapat dinyatakan pada persamaan 2.19 berikut:

$$\min r(w) = \frac{1}{2} \|w\|^2 - \sum_{i=1}^n a_i y_i (w^T x_i + b) - \sum_{i=1}^n a_i \quad (2.19)$$

Dengan memperlihatkan persamaan 2.20 dan 2.21 berikut:

$$w - \sum_{i=1}^n a_i y_i x_i = 0 \quad (2.20)$$

$$\sum_{i=1}^n a_i y_i = 0 \quad (2.21)$$

Metode diatas adalah metode *primal lagrange*, sedangkan dengan memaksimalkan L terhadap a_i , persamaannya 2.22 menjadi persamaan berikut:

$$\max \sum_{i=1}^n a_i - \frac{1}{2} \sum_{i=1, j=1}^n a_i a_j y_i y_j^T x_i x_j^T \quad (2.22)$$

Dengan memperhatikan persamaan 2.23 berikut:

$$\sum_{i=1}^n a_i y_i = 0, a_i \geq 0 \quad (i, j = 1, \dots, n) \quad (2.23)$$

Nilai x didapatkan dari persamaan 2.24 kernel linier untuk x berikut:

$$\sum_{i=1, j=1}^n x_i x_i^T = (i, j = 1 \dots, n) \quad (2.24)$$

Kemudian mencari parameter a_i , dengan terlebih dahulu mencari nilai fungsi setiap abstrak menggunakan persamaan 2.25, lalu mencari nilai a_i , pada persamaan linear menggunakan persamaan 2.26 dengan memperhatikan $i, j = 1, \dots, n$ berikut:

$$\sum_{i=1, j=1}^n a_i S_i^T a_i \quad (2.25)$$

$$\sum_{i=1, j=1}^n a_i S_i^T a_i = y_i \quad (2.26)$$

Setelah parameter a_i didapatkan kemudian dimasukkan ke persamaan 2.28 berikut:

$$\hat{W} = \sum_{i=1}^n a_i S_i \quad (2.27)$$

Hasil yang didapatkan menggunakan persamaan 2.28, selanjutnya digunakan persamaan 2.29, untuk mendapatkan nilai w :

$$y = wx + b \quad (2.28)$$

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

Terdapat empat jenis fungsi kernel yang dapat digunakan, yaitu [14]:

1) Kernel *linier*

$$K(x, x_k) = x_k^T x \quad (2.29)$$

2) Kernel *polynomial*

$$K(x, x_k) = (x_k^T x + 1)^d \quad (2.30)$$

3) Kernel Gaussian (radial basis function, RBF)

$$K(x, x_k) = \exp\{-\|x - x_k\|_2^2\} / \sigma^2 \quad (2.31)$$

4) Kernel *sigmoid*

$$K(x, x_k) = \tanh[\kappa x_k^T x + \theta] \quad (2.32)$$

Hyperplane yang menjadi pemisah terbaik dapat ditemukan dengan mengukur *margin hyperplane* dan mencari titik maksimalnya. Untuk pembahasan suatu kasus yang dapat dipisahkan secara linier, maka dalam hal ini fungsi pemisah yang dicari adalah fungsi linier. Fungsi tersebut dapat didefinisikan sebagai:

$$g(x) = \text{sqn}(f(x)) \quad (2.33)$$

Dengan:

$$f(x) = w^T X + b \quad (2.34)$$

Masalah klasifikasi ini dapat dirumuskan berikut: kita akan menemukan set parameter (w,b) sehingga $f(x) = \langle w, x \rangle + b = y_i$, untuk semua i . Dalam teknik ini kita berusaha menemukan fungsi pemisah (*classifier/hyperplane*) terbaik diantara fungsi yang tidak terbatas jumlahnya untuk memisahkan dua macam objek.

Hyperplane terbaik adalah *hyperplane* yang terletak ditengah-tengah antara dua set objek dari dua kelas. Mencari *hyperplane* terbaik ekuivalen dengan memaksimalkan *margin* atau jarak antara dua set objek dari kelas yang berbeda. Jika $WX_1 + b = +1$ adalah *hyperplane* pendukung dari kelas +1 ($WX_1 + b = +1$) dan $WX_2 + b = -1$ *hyperplane* pendukung dari kelas -1 ($WX_2 + b = -1$), *margin* antara dua kelas dapat dihitung dengan mencari jarak antara kedua *hyperplane-hyperplane* pendukung dari kedua kelas. Secara spesifik *margin* dihitung dengan cara berikut:

$$(WX_1 + b = +1) - (WX_2 + b = -1) = W(X_1 + X_2) = 2 \Rightarrow \quad (2.35)$$

$$LD = \sum_{i=1}^n a_i - \frac{1}{2} \sum_{i,j} a_i a_j y_i y_j (x_i^T x_j + 1)^d \quad (2.36)$$

$$\text{syarat 1 : } \sum_{i=1}^n a_i y_i = 0$$

$$\text{syarat 2 : } a_i \geq 0, i = 1, 2, \dots, N$$

$x_i \cdot x_j$ merupakan *dot product* dua data dalam data latih. *Hyperplane* (batas keputusan atau pemisah).

Maksimasi ini menghasilkan a_i yang bernilai positif. Data-data yang berhubungan dengan a_i positif inilah yang disebut sebagai *support vector* SV, yaitu data-data terluar pada kedua kelas. Setelah mendapatkan *support vector* dengan a_i positif, maka dapat menghitung w dengan persamaan 2.38.

$$w = \sum_{i=1}^n a_i y_i x_i \quad (2.37)$$

Dalam penelitian ini parameter b diberi nilai 0 [17].

Untuk membuktikan bahwa memaksimalkan *margin* antara dua set objek akan meningkatkan probabilitas pengelompokkan secara benar dari data *testing*. Pada dasarnya jumlah fungsi pemisah tidak terbatas jumlahnya, misalnya dari jumlah yang tak terbatas kita ambil dua fungsi, yaitu $f_1(x)$ dan $f_2(x)$. Fungsi f_1 memiliki *margin* yang lebih besar daripada f_2 . Setelah menemukan dua fungsi ini, sekarang suatu data baru masuk dengan keluaran -1 . Maka kita harus mengelompokkan apakah data ini ada dalam kelas -1 atau $+1$ menggunakan fungsi pemisah yang sudah kita temukan. Dengan menggunakan f_1 , kita akan mengelompokkan data baru ini di kelas -1 yang berarti kita benar mengelompokkannya. Kemudian dengan f_2 kita akan menemukannya di kelas $+1$ yang berarti salah. Dari contoh sederhana ini kita lihat bahwa memperbesar *margin* bisa meningkatkan probabilitas pengelompokkan suatu data secara benar.

2.7.1 Multi Class SVM

Pada prinsipnya, *Multi Class SVM* dapat di implementasikan dengan cara menggabungkan beberapa SVM biner. Beberapa metode yang menggunakan cara ini yaitu [14]:

1) One against all

Metode ini mempunyai cara membandingkan satu kelas dengan semua kelas lainnya, sehingga dikenal sebagai *One Against the Rest*. Untuk mengklasifikasikan data kedalam n kelas, sejumlah n model SVM haruslah dibangun. Setiap model SVM ke- i dilatih dahulu menggunakan keseluruhan

data latih, kemudian untuk menjawab apakah sebuah data dapat diklasifikasikan sebagai kelas ke- i atau tidak.

Contoh dalam penggunaan metode ini, untuk mengklasifikasikan data kedalam 4 kelas, maka yang pertama harus dilakukan adalah membangun 4 buah SVM biner. SVM biner pertama dilatih menggunakan keseluruhan data latih untuk mengklasifikasikan data masuk kedalam kelas C1 atau bukan. SVM biner kedua dilatih menggunakan keseluruhan data latih untuk mengklasifikasikan data masuk kedalam kelas C2 atau bukan. Tahap ini dilakukan terus menerus sampai SVM ke empat dilatih menggunakan keseluruhan data untuk mengklasifikasikan data masuk kedalam kelas C4 atau bukan. Tahap pelatihan dengan 4 buah SVM biner dapat dilihat pada Tabel 2.21 dan penggunaannya dalam mengklasifikasi data baru dapat dilihat pada Gambar 2.5.

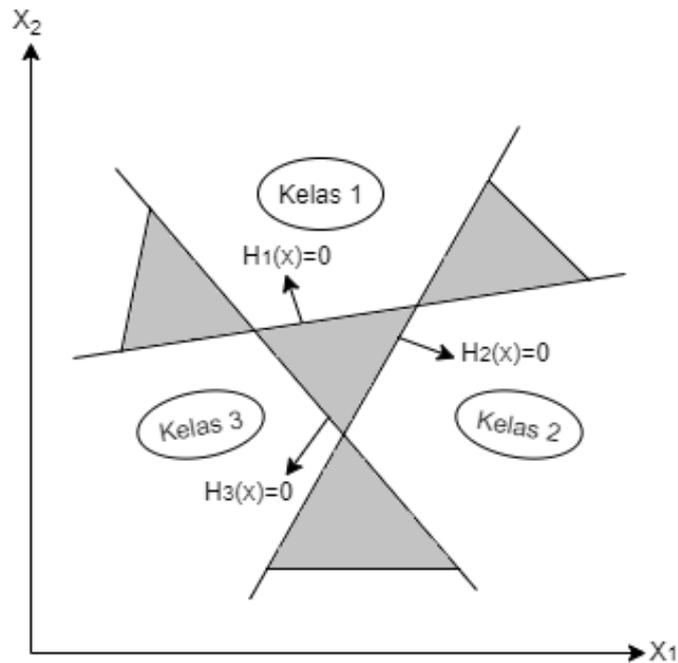
Tabel 2.21 Contoh 4 SVM biner dengan metode *one against all*

$y_i=1$	$y_i=-1$	Hipotesis
Kelas 1	Bukan kelas 1	$f^1(x) = (w^1)x + b^1$
Kelas 2	Bukan kelas 2	$f^2(x) = (w^2)x + b^2$
Kelas 3	Bukan kelas 3	$f^3(x) = (w^3)x + b^3$
Kelas 4	Bukan kelas 4	$f^4(x) = (w^4)x + b^4$

Konsep pada OAA yaitu dimisalkan pada kasus lima kelas, kelas 1, 2, 3, dan 4. Bila akan diujikan $\rho^{(1)}$, semua data dalam kelas 1 diberi label +1 dan data dari kelas lainnya diberi label -1. Pada $\rho^{(2)}$, semua data dalam kelas 2 diberi label +1 dan data dari kelas lainnya diberi label -1 dst hingga data terakhir. Kemudian dicari *hyperplane* dengan algoritma SVM dua kelas. Maka akan didapat *hyperplane* untuk masing-masing kelas diatas. Kemudian kelas dari suatu data baru x ditentukan berdasarkan nilai terbesar dari *hyperplane* [17]:

$$kelas\ x = \arg \max_{l=1, \dots, k} ((w^{(l)})^T \cdot \phi(x) + b^{(l)}) \quad (2.38)$$

Metode *one against all* ini mengakibatkan terbentuknya area-area yang tidak dapat di klasifikasikan (tidak termasuk kelas manapun). area-area yang tidak dapat di klasifikasikan (*unclassifiable regions*) ditunjukkan dengan warna abu abu pada gambar.



Gambar 2.5 Metode OAA menghasilkan area yang tidak dapat di klasifikasikan.

2) *One against one*

Pada penggunaan metode ini, pertama-tama haruslah membuat sejumlah model SVM biner yang membandingkan satu kelas dengan kelas yang lainnya. Untuk melakukan klasifikasi data kedalam k kelas, yang pertama harus dilakukan adalah membangun sejumlah $\frac{k(k-1)}{2}$ model SVM biner. Contoh penerapan SVM dengan klasifikasi 4 kelas, yang pertama adalah membangun $\frac{4(4-1)}{2}=6$ buah SVM biner.

- a) SVM biner pertama dilatih menggunakan data latih kelas pertama atau kelas kedua untuk mengklasifikasi data ke dalam kelas C_1 atau C_2 .
- b) SVM biner kedua dilatih menggunakan data latih dari kelas pertama atau kelas ketiga untuk mengklasifikasikan data kedalam kelas C_1 atau C_3 .

- c) SVM biner ketiga dilatih menggunakan data latih dari kelas pertama dan kelas keempat untuk mengklasifikasikan data kedalam kelas C_1 atau C_4 .
- d) SVM biner keempat dilatih menggunakan data latih dari kelas kedua atau kelas ketiga untuk mengklasifikasikan data kedalam kelas C_2 atau C_3 .
- e) SVM biner kelima dilatih menggunakan data latih dari kelas kedua atau kelas keempat untuk mengklasifikasikan data kedalam kelas C_2 atau C_4 .
- f) SVM biner yang keenam dilatih menggunakan data latih dari kelas ketiga atau keempat untuk mengklasifikasikan data kedalam kelas C_3 atau C_4 .

Pada metode ini setiap kelas harus dibandingkan dengan kelas lainnya, mirip seperti kompetisi penuh dalam pertandingan bola. Untuk mendapatkan kelas keputusan dapat dilakukan dengan menggunakan voting. Jadi, kelas yang paling sering menang adalah kelas keputusan.

Misalnya, jika kelas C_2 menang tiga kali melawan C_1 , C_3 , dan C_4 , kelas C_1 menang dua kali melawan C_3 dan C_4 , dan kelas C_3 menang sekali melawan kelas C_4 , dan kelas C_4 tidak pernah menang sama sekali, maka dapat diambil keputusan bahwa kelas keputusan adalah kelas C_2 . [14] Terdapat beberapa metode untuk melakukan pengujian setelah keseluruhan $\frac{k(k-1)}{2}$ model klasifikasi selesai dibangun. Salah satunya adalah metode voting [18].

Tabel 2.22 Contoh 4 SVM biner dengan metode *one against one*

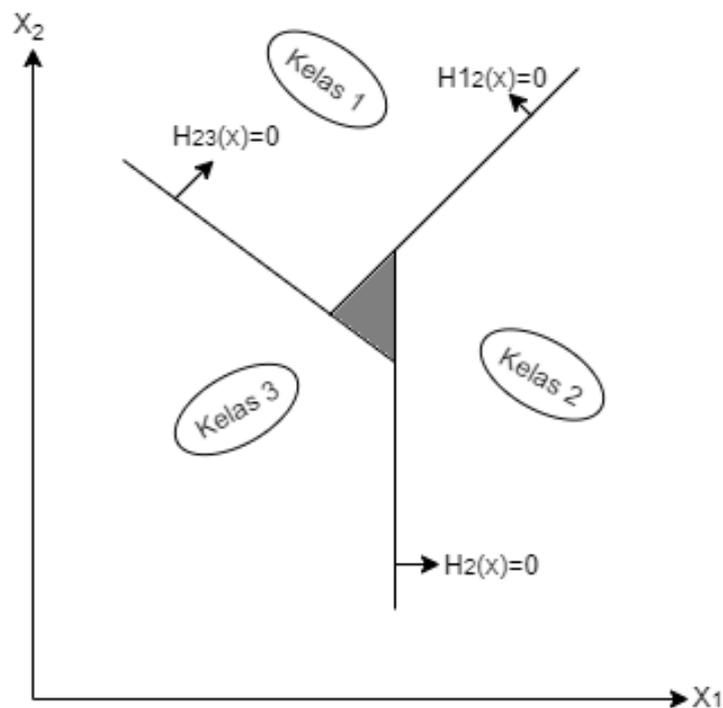
$y_i=1$	$y_i=-1$	Hipotesis
Kelas 1	Kelas 2	$f^{12}(x) = (w^{12})x + b^{12}$
Kelas 1	Kelas 3	$f^{13}(x) = (w^{13})x + b^{13}$
Kelas 1	Kelas 4	$f^{14}(x) = (w^{14})x + b^{14}$
Kelas 2	Kelas 3	$f^{23}(x) = (w^{23})x + b^{23}$
Kelas 2	Kelas 4	$f^{24}(x) = (w^{24})x + b^{24}$
Kelas 3	Kelas 4	$f^{34}(x) = (w^{34})x + b^{34}$

Jika data x dimasukkan kedalam fungsi hasil pelatihan:

$$f(x) = ((w^{ij})^T \phi(x) + b) \quad (2.39)$$

Dan hasilnya menyatakan x adalah kelas i , maka suara untuk kelas i ditambah satu. Kelas dari data x akan ditentukan dari jumlah suara terbanyak. Jika terdapat dua buah kelas yang jumlah suaranya sama, maka kelas yang indeksnya lebih kecil dinyatakan sebagai kelas dari data. Jadi pada pendekatan ini terdapat $\frac{k(k-1)}{2}$ buah persamaan *Quadratic Programming* yang masing-masing memiliki $2n/k$ variabel (n adalah jumlah data pelatihan). [18]

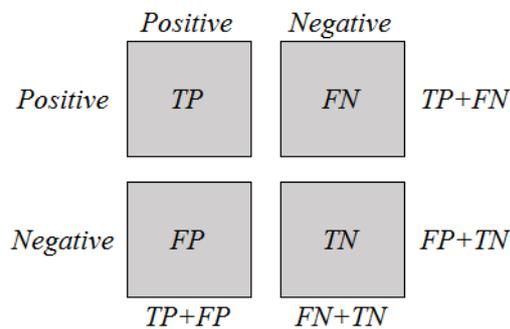
Seperti halnya metode *One against all*, metode *one against one* juga memungkinkan terbentuknya area-area yang tidak dapat diklasifikasikan. Area yang tidak dapat di klasifikasikan ditunjukkan dengan warna abu abu pada gambar.



Gambar 2.6 Metode OAO menghasilkan area yang tidak dapat di klasifikasikan.

2.8 Pengujian Akurasi

Perhitungan akurasi digunakan untuk mengetahui seberapa besar performansi yang ada terhadap hasil pengujian. Dalam penelitian ini, pengujian performansi dilakukan dengan menghitung akurasi dan *F1 score*, sementara untuk mendapatkan *precision* dan *recall* akan menggunakan *Confussion matrix*. *Confussion matrix* adalah salah satu metode yang dipakai untuk menganalisis *tuple* dari data yang berbeda [19].



Gambar 2.7 *Confussion matrix*

- 1) Akurasi merupakan proporsi dari total prediksi *true* semua data. Adapun untuk menghitung akurasi fitur dapat menggunakan persamaan berikut: [20]

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

Keterangan:

TP : *True positive*

FN : *False Negative*

FP : *False Positive*

TN : *True Negative*

- 2) *Precision* merupakan ketepatan dari hasil suatu model. Persamaanya merupakan perbandingan antara *true positive* dengan total data dengan label positif. Adapun untuk menghitung *precision* dapat menggunakan persamaan berikut:

$$\text{Precision} = \frac{TP}{TP + FP} \quad (2.41)$$

- 3) *Recall* adalah ukuran kelengkapan dari suatu model. Persamaannya merupakan perbandingan antara *true positive* dan total contoh yang benar-

benar positif. Adapun untuk menghitung *recall* dapat menggunakan persamaan berikut:

$$Recall = \frac{TP}{TP+FN} \quad (2.42)$$

- 4) *F-Measure* merupakan *harmonic mean* dari *precision* dan *recall*. Adapun untuk menghitungnya dapat menggunakan persamaan berikut:

$$F1\ Score = \frac{2*(Precision*recall)}{Precision+recall} \quad (2.43)$$

Jika dalam suatu pengujian jumlah kelas yang diklasifikasikan lebih dari dua kelas, maka untuk menghitung *precision* dan *recall* dapat menggunakan persamaan berikut [21]:

$$Precision = \frac{\sum_i^L \frac{TP_i}{TP_i+FP_i}}{L} * 100\% \quad (2.44)$$

$$Recall = \frac{\sum_i^L \frac{TP_i}{TP_i+FN_i}}{L} * 100\% \quad (2.45)$$

Keterangan:

L :Jumlah keseluruhan data yang di uji/jumlah kelas

2.10 Pengertian Python

Python adalah bahasa pemrograman komputer, sama layaknya seperti bahasa pemrograman lain, misalkan Pascal,C,C++, Java, PHP dan lain-lain. sebagai bahasa pemrograman, python tentu memiliki dialek, kosakata atau kata kunci (*keyword*) dan aturan tersendiri yang jelas berbeda dengan bahasa pemrograman lainnya. Bahasa pemrograman python disusun pada akhir tahun 1980-an dan baru di implementasikan pada Desember 1989 oleh Guido Van Rossum di Centrum Wiskunde & Informatica (CWI), sebuah riset pada bidang matematika dan sains, Amsterdam-Belanda. Sebagai suksesor atas pengganti bahasa pemrograman sebelumnya, bahasa ABC, yang juga dikembangkan di CWI oleh Leo Guerts, Lambert Martens, dan Steven Pemberton.

2.11 Flowchart

Bagan alir atau *flowchart* merupakan bagan-bagan yang mempunyai arus yang menggambarkan langkah-langkah penyelesaian suatu masalah secara sederhana, teratur, rapi, dan jelas. *Flowchart* juga merupakan cara penyajian dari suatu algoritma [22]. *Flowchart* menggambarkan langkah penyelesaian masalah

dengan merepresentasikan simbol-simbol tertentu yang mudah dimengerti, standar dan mudah digunakan (Ladjamudin, 2005).

Berikut adalah pedoman pada saat akan membuat *flowchart*:

- 1) *Flowchart* sebaiknya digambarkan dari atas ke bawah dan mulai dari bagian kiri suatu halaman.
- 2) Di dalam *flowchart* kumpulan kegiatan haruslah ditunjukkan dengan jelas.
- 3) Dimulainya kegiatan, dan berakhirnya kegiatan harus ditunjukkan dengan jelas.
- 4) Masing-masing kegiatan di dalam *flowchart* sebaiknya menggunakan kata yang mewakili suatu pekerjaan.
- 5) Semua kegiatan di dalam *flowchart* harus diurutkan dengan semestinya.
- 6) Kegiatan yang terpotong dan akan disambung di tempat lain harus ditunjukkan dengan jelas oleh simbol penghubung.
- 7) Menggunakan simbol-simbol standar untuk *flowchart*.

2.12 Diagram Konteks

Context diagram atau diagram konteks adalah tingkatan tertinggi dalam diagram aliran data, dikarenakan dalam diagram ini hanya memuat satu proses, Dalam diagram konteks hanya ada satu proses yang menggambarkan sistem secara keseluruhan dan tidak boleh terdapat *store*, sistem juga dibatasi oleh *boundary* (dapat digambarkan dengan garis putus). Proses tersebut diberi nomor nol, didalamnya terdapat semua entitas eksternal yang ditunjukkan pada diagram konteks berikut aliran data-data utama menuju dari sistem. Diagram konteks merupakan gambaran awal yang mencakup dasar, sistem, umum, dan keluaran. (Kendall dan Kendall, 2003).

2.13 Data Flow Diagram (DFD)

Diagram alir data atau *Data Flow Diagram* (DFD) merupakan sebuah model dari sistem untuk menggambarkan pembagian sistem ke dalam sebuah modul yang lebih kecil. Penggunaan DFD ini mempunyai banyak keuntungan, salah satunya adalah memudahkan pemakai atau user yang kurang menguasai komputer untuk mengerti maksud sistem yang sedang dikerjakan (Ladjamudin, 2005). Karena DFD memberikan pandangan secara menyeluruh mengenai sistem yang ditangani, menunjukkan fungsi-fungsi utama atau proses yang ada, aliran data, dan *external entity*.

2.14 Pengujian Black Box

Pengujian *black box* dilakukan untuk menemukan kesalahan serta kekurangan yang tidak dapat ditemukan sebelumnya, pengujian ini dikhususkan untuk menguji fungsionalitas serta kinerja sistem yang telah dibangun.