

SKRIPSI

KLASIFIKASI TINGKAT KEMATANGAN BUAH KOPI

Disusun dan diajukan oleh:

**PUTRI ANGRANI
D421 16 506**



**PROGRAM STUDI SARJANA TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS HASANUDDIN
GOWA
2023**

LEMBAR PENGESAHAN SKRIPSI
KLASIFIKASI TINGKAT KEMATANGAN BUAH KOPI

Disusun dan diajukan oleh

PUTRI ANGRANI

D42116506

Telah dipertahankan di hadapan Panitia Ujian yang dibentuk dalam rangka
Penyelesaian Studi Program Sarjana Program Studi Teknik Informatika Fakultas
Teknik Universitas Hasanuddin pada tanggal 12 Juli 2023 dan dinyatakan telah
memenuhi syarat kelulusan.

Menyetujui,

Pembimbing Utama,

Pembimbing Pendamping,

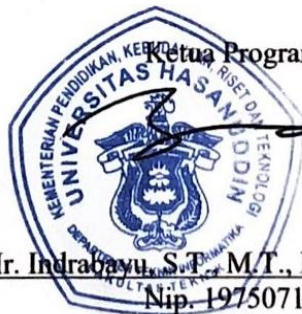


Prof. Dr. Ir. Indrabayu, S.T., M.T.,
M.Bus.Sys., IPM., ASEAN.Eng.
Nip. 197507162002121004



Ir. Christoforus Yohannes, MT.
Nip. 196007161987021002

Ketua Program Studi,



Prof. Dr. Ir. Indrabayu, S.T., M.T., M.Bus.Sys., IPM., ASEAN.Eng.
Nip. 197507162002121004

PERNYATAAN KEASLIAN

Yang bertanda tangan dibawah ini ;

Nama : Putri Angriani
NIM : D42116506
Program Studi : Teknik Informatika
Jenjang : S1

Menyatakan dengan ini bahwa karya tulisan saya berjudul

KLASIFIKASI TINGKAT KEMATANGAN BUAH KOPI

Adalah karya tulisan saya sendiri dan bukan merupakan pengambilan alihan tulisan orang lain dan bahwa skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri.

Semua informasi yang ditulis dalam skripsi yang berasal dari penulis lain telah diberi penghargaan, yakni dengan mengutip sumber dan tahun penerbitannya. Oleh karena itu semua tulisan dalam skripsi ini sepenuhnya menjadi tanggung jawab penulis. Apabila ada pihak manapun yang merasa ada kesamaan judul dan atau hasil temuan dalam skripsi ini, maka penulis siap untuk diklarifikasi dan mempertanggungjawabkan segala resiko.

Segala data dan informasi yang diperoleh selama proses pembuatan skripsi, yang akan dipublikasi oleh Penulis di masa depan harus mendapat persetujuan dari Dosen Pembimbing.

Apabila dikemudian hari terbukti atau dapat dibuktikan bahwa sebagian atau keseluruhan isi skripsi ini hasil karya orang lain, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Gowa, 8 Juni 2023

Yang Menyatakan



Putri Angriani

ABSTRAK

PUTRI ANGRANI. *Klasifikasi Tingkat Kematangan Buah Kopi* (dibimbing oleh Prof. Dr. Ir. Indrabayu., ST, MT, M.Bus.Sys., IPM, ASEAN. Eng. dan Ir. Christoforus Yohannes, M.T.)

Pabrik kopi memiliki standar tingkat kematangan buah kopi agar kopi yang dijual konsisten dari segi harga maupun rasa. Pabrik-pabrik kopi sangat mengedepankan kualitas. Tidak semua buah yang didapat dari petani maupun pengepul memiliki kualitas yang baik dari segi kematangannya. Pemilahan buah kopi secara manual memiliki banyak kendala, baik dari adanya sifat subjektif, memakan waktu yang cukup lama, serta melelahkan. Oleh karena itu, penelitian ini bertujuan untuk membuat sistem yang dapat mensegmentasi serta mengklasifikasi tingkat kematangan buah kopi menggunakan metode SVM. Klasifikasi tingkat kematangan buah kopi dibagi menjadi tiga kelas yakni muda, mengkal, dan masak. Data yang digunakan berupa video. Video untuk tahap *training* pada masing-masing kelas memiliki 30 buah kopi. Sedangkan pada tahap *testing* memiliki total 42 buah kopi yang tercampur dan direkam secara acak pada setiap kelasnya. Kemudian, video diekstrak menjadi sekumpulan gambar yang memiliki resolusi 1920 x 1080 piksel dan warna RGB. Metode yang digunakan pada segmentasi yaitu mengkonversi RGB ke HSV, *masking* gambar, deteksi tepi, dan operasi morfologi menggunakan operasi erosi. Adapun metode klasifikasi pada tingkat kematangan buah kopi yaitu *Support Vector Machine* (SVM). Parameter yang digunakan adalah kernel RBF, $cost = 100$, dan $gamma = 0.1$ yang diperoleh dari *Grid Search*, serta multi kelas OAA (*One Against All*). Hasil akurasi pada klasifikasi tingkat kematangan buah kopi menggunakan algoritma SVM yaitu sebesar 96%.

Kata kunci: Kopi, HSV, SVM, Multi Kelas OAA

ABSTRACT

PUTRI ANGRIANI, *Classification of the Maturity Levels of Coffee* (supervised by Prof. Dr. Ir. Indrabayu., ST, MT, M.Bus.Sys., IPM, ASEAN. Eng. and Ir. Christoforus Yohannes, M.T.)

The coffee factory has a standard level of ripeness for the coffee fruit so that the coffee sold is consistent in terms of price and taste. Coffee factories put quality first. Not all fruits obtained from farmers and collectors have good quality in terms of maturity. Sorting coffee cherries manually has many obstacles, both from a subjective nature takes quite a long time and is tiring. Therefore, this study aims to create a system that can segment and classify coffee cherries using the SVM method. The classification of the maturity level of the coffee cherries is divided into three classes namely young, unripe, and ripe. The data used is in the form of a video. The videos for the training phase in each category have 30 copies. In the testing phase, 42 coffees were mixed and recorded randomly in each category. Then, the video is extracted into a set of images with a resolution of 1920 x 1080 pixels and RGB colors. The method used in segmentation is converting RGB to HSV, image masking, edge detection, and morphological operations using erosion operations. The classification method for the maturity level of the coffee cherries is Support Vector Machine (SVM). The parameters used are the RBF kernel, cost = 100, gamma = 0.1 obtained from Grid Search, and multi-class OAA (One Against All). The results of the accuracy in the classification of the maturity level of the coffee cherries using the SVM algorithm are 96%.

Keywords: Coffee, HSV, SVM, multi-class OAA

DAFTAR ISI

LEMBAR PENGESAHAN SKRIPSI.....	i
PERNYATAAN KEASLIAN.....	ii
ABSTRAK	iii
ABSTRACT	iv
DAFTAR ISI.....	v
DAFTAR GAMBAR	vii
DAFTAR TABEL.....	viii
DAFTAR SINGKATAN DAN ARTI SIMBOL	ix
DAFTAR LAMPIRAN.....	x
KATA PENGANTAR	xi
BAB I PENDAHULUAN	1
1.1. Latar Belakang.....	1
1.2. Rumusan Masalah	2
1.3. Tujuan Penelitian.....	2
1.4. Manfaat Penelitian.....	2
1.5. Ruang Lingkup	2
BAB II TINJAUAN PUSTAKA.....	4
2.1. Kopi	4
2.2. Pengolahan Citra Digital	5
2.3. Visi Komputer	9
2.4. Gaussian Filter.....	11
2.5. RGB Color Channel	12
2.6. Segmentasi Warna menggunakan <i>Hue, Saturation, dan Value</i> (HSV) ...	12
2.7. Thresholding.....	14
2.8. Blob Detection.....	15
2.9. Support Vector Machine (SVM)	16
2.10. Multi Kelas SVM	22
BAB III METODE PENELITIAN.....	26

3.1	Lokasi Penelitian	26
3.2	Tahapan Penelitian	26
3.3.	Instrumen Penelitian.....	27
3.4	Teknik Pengambilan Data	28
3.5	Perancangan Sistem.....	29
3.6.	Analisis Kerja Sistem	38
BAB IV HASIL DAN PEMBAHASAN		40
4. 1	Hasil Penelitian.....	40
4. 2	Hasil Pengujian Sistem.....	54
4. 3	Pembahasan	56
BAB V KESIMPULAN DAN SARAN.....		60
5.1.	Kesimpulan.....	60
5.2	Saran	60
DAFTAR PUSTAKA		61
LAMPIRAN		65

DAFTAR GAMBAR

Gambar 1 Perubahan Fisik Buah Kopi Selama Proses Kematangan di Pohon	4
Gambar 2 Ruang Warna HSV	14
Gambar 3 Area Blob	15
Gambar 4 Ilustrasi Pencarian Hyperplane Terbaik Menggunakan SVM.....	16
Gambar 5 Beberapa Misklasifikasi Pada Soft Margin SVM	18
Gambar 6 Kernel SVM Untuk Memisahkan Data Secara Linear	20
Gambar 7 Contoh klasifikasi dengan metode One-Against-All (Permata, 2016).	23
Gambar 8 Contoh klasifikasi dengan metode One-Against-One (Pricila, 2016).	25
Gambar 9 Lokasi Penelitian	26
Gambar 10 Tahap Penelitian.....	26
Gambar 11 Skenario Pengambilan Data	29
Gambar 12 Flowchart Perancangan Sistem	30
Gambar 13 Alur Sistem Proses Pelatihan SVM.....	35
Gambar 14 Frame Buah Kopi Hasil Klasifikasi.....	38
Gambar 15 Frame Buah Kopi Pada Tahap Training.....	40
Gambar 16 Tahap Preprocessing.....	43
Gambar 17 Blurring Gambar	43
Gambar 18 Contoh Konversi dari RGB ke HSV	44
Gambar 19 Nilai matriks konversi gambar HSV ke gambar biner	45
Gambar 20 Hasil Proses Masking	46
Gambar 21 Model training SVM	48
Gambar 22 Frame pada Video	50
Gambar 23 Preprocessing	51
Gambar 24 Hasil gabungan dari kedua gambar biner	52
Gambar 25 Object Detection.....	52
Gambar 26 Pengambilan Objek untuk Blob Detection.....	53
Gambar 27 Frame Buah Kopi Hasil Klasifikasi.....	54
Gambar 28 Hasil Prediksi Kematangan Buah Kopi.....	55

DAFTAR TABEL

Tabel 1	Contoh 4 SVM biner dengan metode One-Against-All.....	23
Tabel 2	Contoh 6 SVM biner dengan metode One-Against-One	25
Tabel 3	Kriteria Buah Kopi.....	31
Tabel 4	Perbandingan Ukuran Gambar.....	41
Tabel 5	Kombinasi nilai parameter alpha dan beta optimal.....	41
Tabel 6	Range Nilai HSV	45
Tabel 7	Rata-rata Nilai RGB.....	46
Tabel 8	Nilai Bias.....	47
Tabel 9	Support Vector untuk Setiap Kelas.....	48
Tabel 10	Hasil Akurasi Sistem Menggunakan SVM	55
Tabel 11	Kesalahan pada Proses Segmentasi Buah Kopi	56
Tabel 12	Beberapa Kesalahan Hasil Klasifikasi	57
Tabel 13	Contoh Hasil Kesalahan Klasifikasi	58

DAFTAR SINGKATAN DAN ARTI SIMBOL

Lambang/Singkatan	Arti dan Keterangan
σ	nilai <i>standard deviation</i>
π	konstanta dengan nilai 3.14
e	konstanta bilangan natural dengan nilai 2, 71828182
μ	nilai rata-rata (mean)
$M \times N$	ukuran citra
P	piksel citra
H	<i>Hue</i> (warna sebenarnya)
S	<i>Saturation</i> (kemurnian warna)
V	<i>Value</i> (kecerahan)
T	Threshold nilai ambang
B	Bias
C	Cost
γ	gamma

DAFTAR LAMPIRAN

Lampiran 1. Source Code.....	65
Lampiran 2. Dataset	65

KATA PENGANTAR

Assalamu'alaikum Warahmatullaahi Wabarakatuh.

Alhamdulillahirabbil'aalamiin, puji syukur senantiasa penulis panjatkan atas kehadiran Allah SWT yang telah melimpahkan rahmat dan hidayah-Nya yang sehingga penulis dapat menyelesaikan laporan Tugas Akhir ini yang berjudul "Klasifikasi Tingkat Kematangan Buah Kopi" sebagai salah satu syarat untuk memperoleh gelar Sarjana Strata Satu (S1) pada Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin. Sholawat serta salam InsyaAllah selalu tercurahkan kepada junjungan kita Nabi Muhammad SAW yang telah menunjukkan dan mengajarkan akhlak mulia sehingga didapatkan kenyamanan.

Dalam proses pembuatan laporan akhir ini tidak lepas dari bantuan, bimbingan serta dukungan dari berbagai pihak, dari masa perkuliahan sampai dengan masa penyusunan tugas akhir ini. Oleh karena itu, penulis dengan senang mengucapkan rasa terima kasih kepada:

1. Kepada Orang Tua dan keluarga, Bapak Jumardi dan Ibu Darna yang selalu memberikan dukungan, motivasi, dan semangat serta selalu sabar menunggu ku hingga sarjana, tidak henti-hentinya mendoakan, merawat, menyayangi, terima kasih. ♥
2. Bapak Prof. Dr. Ir. Indrabayu., ST, MT, M.Bus.Sys., IPM, ASEAN. Eng., selaku pembimbing I dan Bapak Ir. Christoforus Yohannes, M.T., selaku pembimbing II yang selalu menyediakan waktu, tenaga, dan pikirannya yang luar biasa untuk mengarahkan penulis dalam penyusunan tugas akhir ini.

3. Bapak Prof. Dr. Ir. Indrabayu., ST, MT, M.Bus.Sys., IPM, ASEAN. Eng., selaku Ketua Departemen Teknik Informatika dan selaku Kepala Lab *Artificial Intelligence* Teknik Informatika Fakultas Teknik Univeristas Hasanuddin atas bimbingannya selama masa perkuliahan.
4. Segenap Dosen dan Staff Departemen Teknik Informatika Fakultas Teknik Univesitas Hasanuddin yang telah banyak membantu dan memberikan banyak ilmu selama masa perkuliahan.
5. Sahabat-sahabat A I R O K U, Andi Marimar Muchtamar, Cici Purnamasari, Dhinda Fitri Wiludjeng, Sitti Nur Fadillah dan Ghina Syukriyah yang selalu menjadi tempat berbagi cerita, memberikan perhatian, dukungan dan semangat.
6. Para komplotan anak-anak Sudiang, Daya, Antang, Gowa yang selalu membersamai hari-hari ku dan saudara-saudari I G N I T E R 16 atas dukungan dan semangat yang diberikan selama ini.
7. Sahabat-sahabat A G P, Firdayanti Allolinggi', Ayu Trisnawati, Genoria Lestari Nasran, Glorya Lebang dan Aprilia Pute' yang menjadi tempat berbagi cerita dan meluangkan waktu, memberikan perhatian, dukungan dan semangat.
8. Kak Nur Hikmah yang telah meminjamkan *conveyor belt* yang digunakan dalam pengambilan data penelitian.
9. Teman-teman dan kakak-kakak AIMP *Research Group* FT-UH yang telah memberikan bantuan selama penelitian dan diskusi terkait progres penyusunan Tugas Akhir.

10. Orang-orang berpengaruh lainnya yang tidak sempat penulis sebutkan satu persatu yang telah memberikan dukungan, semangat dan doanya selama penyusunan tugas akhir ini.

Akhirnya dengan segala kerendahan hati, penyusun menyadari bahwa masih terdapat kekurangan dalam penyusunan Tugas Akhir ini baik isi maupun cara penyajian. Oleh karena itu penyusun mengharapkan adanya saran dan kritik yang bersifat membangun demi kesempurnaan tugas akhir ini. Penulis berharap semoga tugas akhir ini dapat memberikan manfaat bagi pembaca pada umumnya dan manfaat bagi penulis khususnya.

Wassalamu'alaikum Warahmatullaahi Wabarakaatuh

Makassar, Juni 2023

Penulis

BAB I

PENDAHULUAN

1.1. Latar Belakang

Kopi merupakan jenis minuman yang berasal dari proses pengolahan dan ekstraksi biji tanaman kopi, kata kopi sendiri berasal dari bahasa Arab *qahwah* yang berarti kekuatan karena pada awalnya kopi digunakan sebagai makanan berenergi tinggi. Kata *qahwah* mengalami perubahan menjadi *kahveh* berasal dari bahasa Turki dan berubah lagi menjadi *koffee* dalam bahasa Belanda. Penggunaan kata *koffee* segera diserap kedalam Bahasa Indonesia menjadi kata kopi yang dikenal saat ini (Amando, 2020). Dua jenis pohon kopi yang dikenal secara umum yaitu kopi robusta (*Coffea canephora*) dan kopi Arabika (*Coffea arabica*). Kopi arabika tumbuh pada ketinggian 600–2000 m di atas permukaan laut. Tanaman ini dapat tumbuh hingga 3 meter bila kondisi lingkungannya baik. Suhu tumbuh optimalnya adalah 18-26°C. Biji kopi yang dihasilkan berukuran cukup kecil dan berwarna hijau hingga merah gelap (Wulandari, 2021).

Kopi merupakan salah satu komoditas ekspor yang menghasilkan devisa cukup tinggi bagi Indonesia. Untuk peningkatan nilai mutu perlu adanya kopi yang berkualitas baik. Konsumsi kopi dunia mencapai 70% berasal dari spesies kopi arabika dan 26% berasal dari spesies kopi robusta. Pudji Rahardjo dalam bukunya yang berjudul Kopi menjelaskan bahwa saat penyebaran kopi melalui saudagar Arab, mereka mencoba memakan buah kopi dan merasakan adanya tambahan energi yang kemudian buah kopi dimanfaatkan menjadi minuman kopi seperti sekarang ini. Masyarakat di Arab menyebut minuman ini sebagai *bahwa* yang berarti pencegah rasa kantuk karena kopi memiliki kandungan kafein yang tinggi.

Kualitas buah kopi dapat ditentukan pada saat proses panen terjadi. Untuk meningkatkan mutu dan nilai ekspor di dunia tentunya membutuhkan adanya kopi yang berkualitas baik. Tingkat kematangan buah kopi sangat menentukan kualitas kopi tersebut. Buah yang dipanen dengan kondisi buah yang matang sempurna dapat menghasilkan kualitas kopi yang sangat baik apabila diolah dengan benar. Menurut (Ratmawati, 2019), ciri ciri buah kopi yang telah matang bisa dilihat dari warna kulitnya. Buah kopi yang paling baik untuk dipanen adalah yang telah matang penuh, berwarna merah. Namun karena cara panen petani yang masih konvensional, buah yang dipanen tidak sepenuhnya merah. Pabrik kopi memiliki standar tingkat kematangan buah kopi agar kopi yang dijual konsisten dari

segi harga maupun rasa. Pabrik-pabrik kopi sangat mengedepankan kualitas. Tidak semua buah yang didapat dari petani maupun pengepul memiliki kualitas yang baik dari segi kematangannya. Pemilahan buah kopi secara manual memiliki banyak kendala, baik dari adanya sifat subjektif, memakan waktu yang cukup lama, serta melelahkan. Berdasarkan hal tersebut, maka dibutuhkan sistem yang dapat membantu para petani dalam proses penyortiran sehingga dapat mengklasifikasi buah kopi sesuai dengan tingkat kematangan agar petani dapat memperoleh buah kopi yang berkualitas baik.

1.2. Rumusan Masalah

Berdasarkan latar belakang masalah di atas, maka dapat dirumuskan permasalahan pada penelitian ini yaitu:

1. Bagaimana sistem dapat mensegmentasi tingkat kematangan buah kopi?
2. Bagaimana hasil *testing* model dari sistem klasifikasi tingkat kematangan buah kopi menggunakan metode SVM berbasis *video processing*?

1.3. Tujuan Penelitian

Berdasarkan rumusan masalah di atas, maka dapat dibuat tujuan antara lain:

1. Membuat sistem yang dapat mensegmentasi buah kopi.
2. Mengetahui hasil *testing* model dari sistem klasifikasi tingkat kematangan buah kopi menggunakan metode SVM berbasis *video processing*.

1.4. Manfaat Penelitian

Dengan dilakukannya penelitian ini, maka manfaat yang diperoleh dari penelitian ini adalah:

1. Membantu industri kecil menengah dalam upaya peningkatan *quality control*.
2. Mendorong penggunaan teknologi *video processing* pada bidang industri, khususnya pada industri pengolah kopi.

1.5. Ruang Lingkup

Yang menjadi batasan masalah dari penelitian ini adalah sebagai berikut:

1. Buah kopi yang digunakan sebagai data adalah buah kopi Arabika dari daerah Kabupaten Toraja Utara.

2. Objek penelitian berupa buah kopi yang dibagi menjadi 3 kategori berdasarkan warna yaitu muda, mengkal, dan masak.
3. Sistem klasifikasi tingkat kematangan buah kopi menggunakan metode *Support Vector Machine* (SVM) berbasis *video processing*.
4. *Background* citra buah kopi berwarna putih.
5. Kamera yang digunakan yaitu *Logitech Webcam C922 Pro*.
6. Pengambilan data dilakukan dengan pencahayaan yang telah ditentukan.
7. Jarak antara kamera dan objek buah kopi yaitu 25 cm.
8. Data sampel yang digunakan untuk kematangan buah kopi diambil dari satu sisi yaitu bagian atas buah kopi.

BAB II TINJAUAN PUSTAKA

2.1. Kopi

Kopi adalah tanaman perkebunan yang sudah lama dibudidayakan di Indonesia. Salah satu jenis kopi yang ditanam di Indonesia, yaitu kopi arabika. Kopi arabika merupakan kopi yang memiliki kualitas superior dibanding jenis kopi lainnya. Buah kopi arabika diolah dari penjemuran, pengeriangan, dan penyangraian serta penggilingan menjadi kopi bubuk arabika (Srikandi dkk, 2019). Tingkat kematangan buah kopi ditandai dengan warna kulit buah kopi, kematangan buah kopi bervariasi saat masih dipohon mulai dari hijau, hijau kekuningan, kuning, merah kekuningan, merah sempurna dan merah tua kehitaman.

Menurut (Mulato, 2019), daging buah kopi matang terdiri atas campuran bahan organik dan air dalam proporsi tertentu yang layak digunakan sebagai substrat fermentasi. Kadar senyawa gula dalam daging buah kopi sangat tergantung pada tingkat kematangannya. Ukuran kematangan buah ditandai oleh perubahan warna kulit buah yang semula berwarna hijau menjadi merah. Senyawa khlorofil adalah pigmen warna hijau saat buah kopi masih mentah. Pigmen karotenoid menjadi dominan ketika buah kopi sudah matang. Warna ini dipakai acuan untuk menentukan saat yang tepat untuk panen buah kopi. Setelah lewat matang, warna kulit buah berubah menjadi merah kehitaman. Adapun perubahan fisik buah kopi selama proses kematangan di pohon dapat dilihat pada Gambar 1.

Kematangan	Days to harvest, N	Brix	Karakter citarasa
 Mentah	15	---	Buah sangat muda = warna hijau dominan. Tidak boleh dipanen. Karakter citarasa: acidity dan body lemah. Grassy, bitterness, astringency sangat tinggi
 ½-matang	13	12	Buah setengah masak = warna kuning-kemerahan. Tidak disarankan dipanen dulu. Karakter citarasa: acidity, body, dan bitterness sedang.
 Tepat matang	10	14	Buah warna merah merata. Ideal dipanen. Citarasa: optimal, bright acidity, body berat.
 Lewat matang	6	17	Buah warna merah-kehitaman. Kadang mongering. Kelewat masak segera dipanen. Citarasa: mulai menurun. Cacat citarasa: mouldy, stink, earthy.

Gambar 1 Perubahan Fisik Buah Kopi Selama Proses Kematangan di Pohon

(Sumber: www.cctcid.com)

Selain warna, kematangan buah kopi juga dapat diukur dari tingkat kekerasan kulit buahnya. Buah mentah mempunyai kulit yang keras dan sulit dikupas. Diperlukan daya yang besar untuk mengupas buah kopi mentah, yaitu 15 N (Newton). Daya ini akan berkurang menjadi tinggal 10 N saat buah sudah tepat matang. Kulit dan daging buah matang sersifat sangat lunak, berlendir, dan mudah dikupas. Secara kimiawi, kandungan senyawa gula dalam daging buah juga meningkat secara signifikan. Pada buah kopi mentah, kadar gula belum bisa diukur. Seiring dengan proses pematangan buah kopi, kadar gula naik secara bertahap dan mencapai puncaknya sampai 17 Brix. Satuan Brix adalah proporsi antara berat padatan terlarut (termasuk gula) dalam 100 gr larutan. Senyawa gula dalam daging buah berkorelasi positif dengan jenis dan jumlah senyawa pembentuk citarasa dalam biji kopi. Buah kopi merah akan memiliki komposisi kimia senyawa pembentuk citarasa yang lengkap dan maksimal. Selain itu, petik buah kopi merah juga memberikan keuntungan teknis, yaitu mudah dikupas kulitnya tanpa merusak kulit tanduk dan keuntungan ekonomis dalam bentuk peningkatan rendemen olah yang tinggi (Mulato, 2019).

Kopi merupakan tumbuhan yang berbuah kecil bulat dengan ukuran maksimal sebesar biji kelereng. Tumbuhan kopi ini seringkali di buat bubuk, dengan tujuan diminum, dibuat makanan atau keperluan lainnya. Untuk asal istilah kopi menurut banyak ahli berasal dari Bahasa *Gahwah* yaitu dari Bahasa Arab (Wijo, 2015). Kopi ini menjadi salah satu tanaman yang banyak ditemukan di Indonesia, hampir disetiap wilayah Indonesia ada jenis-jenis kopi yang berkualitas. Baik Aceh, Lampung, Sulawesi, dan Kopi di daerah lainnya. Tanaman kopi sudah mulai berbuah pada umur 2,5-3 tahun untuk robusta dan 3-4 tahun untuk arabika. Namun buah kopi pertama biasanya hanya berbuah sedikit. Produktivitasnya mulai naik maksimal setelah berumur 5 tahun ke atas (Wijo, 2015).

2.2. Pengolahan Citra Digital

Pengolahan citra digital adalah sebuah disiplin ilmu yang mempelajari tentang teknik-teknik mengolah citra. Pengolahan citra digital menunjuk pada pemrosesan gambar dua dimensi menggunakan komputer. Dalam konteks yang lebih luas, pengolahan citra digital mengacu pada pemrosesan setiap data 2 dimensi. Pengolahan citra bertujuan untuk memproses citra menjadi citra yang kualitasnya lebih baik. Perbaikan atau modifikasi citra perlu dilakukan untuk meningkatkan kualitas atau menonjolkan beberapa aspek informasi yang terkandung di dalam citra (Shafira, 2018).

Langkah-langkah dalam pengolahan citra digital antara lain adalah:

1. Akuisisi citra, yaitu merupakan proses untuk menangkap atau mengambil citra yang dibutuhkan dengan menggunakan sensor pencitraan berupa kamera, *scanner*, dan lainnya.
2. *Preprocessing* citra, yaitu dilakukan untuk menyiapkan citra untuk diproses lebih lanjut, bisa berupa ekstraksi fitur maupun kebutuhan klasifikasi. Teknik *preprocessing* citra yang umum digunakan antara lain adalah *cropping* dan perubahan ukuran citra.
3. Segmentasi, yaitu membagi sebuah citra menjadi beberapa bagian penyusunnya. Proses segmentasi dilakukan sampai objek yang diinginkan dalam suatu aplikasi terpisah dari objek aslinya. Tingkat kesuksesan dari sebuah sistem pengenalan citra juga dipengaruhi oleh segmentasi yang akurat.
4. Representasi dan deskripsi. Representasi yaitu menyatakan data *pixel* ke dalam bentuk data yang mampu diolah oleh komputer. Sementara proses deksripsi dilakukan untuk mengekstrak atribut yang menghasilkan informasi kuantitatif yang diinginkan atau yang merupakan fitur untuk membedakan citra antar kelas.
5. Pengenalan, yaitu proses pemberian label pada objek sesuai dengan fitur yang dimiliki objek.

Contoh pengolahan citra digital yang sering dilakukan adalah sebagai berikut (Sagala, 2009).

a. Perbaikan Kualitas Citra (*Image Enhancement*)

Image enhancement digunakan untuk memperbaiki kualitas citra dengan memanipulasi parameter-parameter citra sehingga citra yang dihasilkan akan semakin baik. Operasi perbaikan citra adalah Perbaikan kontras gelap/terang, Perbaikan tepian objek (*edge enhancement*), Penajaman (*sharpening*), Pemberian warna semu (*pseudocoloring*), Penapisan derau (*noise filtering*).

b. Pemugaran Citra (*Image Restoration*)

Image restoration digunakan untuk menghilangkan cacat pada citra. Hal ini hampir sama dengan perbaikan citra perbedaannya adalah penyebab degradasi citra diketahui. Operasi pemugaran citra adalah Penghilangan kesamaran (*image deblurring*) dan Penghilangan derau (*noise*).

c. Pemampatan Citra (*Image Compression*)

Image compression digunakan agar citra direpresentasikan dalam bentuk lebih kompak, sehingga keperluan memori lebih sedikit namun dengan tetap mempertahankan kualitas citra.

d. Segmentasi Citra (*Image Segmentation*)

Image segmentation digunakan untuk memecah citra menjadi beberapa *segment* dengan kriteria tertentu. Biasanya berkaitan dengan pengenalan pola. Segmentasi membagi suatu citra menjadi wilayah-wilayah yang homogen berdasarkan kriteria keserupaan tertentu antara derajat keabuan suatu *pixel* dengan derajat keabuan *pixel-pixel* tetangganya.

e. Analisis Citra (*Image Analysis*)

Image analysis digunakan untuk menghitung besaran kuantitatif dari citra untuk menghasilkan deskripsinya. Proses ini biasanya diperlukan untuk melokalisasi objek yang diinginkan dari sekelilingnya. Contoh *image analysis* yaitu Pendeteksian tepian objek (*edge detection*), Representasi daerah (*region*), dan Ekstraksi Fitur. Ekstraksi fitur merupakan bagian fundamental dari analisis citra. Fitur adalah karakteristik unik dari suatu objek. Karakteristik fitur yang baik sebisa mungkin memenuhi persyaratan berikut.

- a. Dapat membedakan suatu objek dengan lainnya.
- b. Memperhatikan kompleksitas komputasi dalam memperoleh fitur.
- c. Tidak terikat dalam arti bersifat invarian terhadap berbagai transformasi (rotasi, penskalaan, dan pergeseran).
- d. Jumlahnya sedikit, karena fitur yang jumlahnya sedikit akan menghemat waktu komputasi dan ruang penyimpanan untuk proses selanjutnya (proses pemanfaatan fitur) (Putra, 2009).

f. Rekontruksi Citra (*Image Reconstruction*)

Image reconstruction digunakan untuk membentuk ulang objek dari beberapa citra hasil proyeksi. Operasi rekontruksi citra biasanya banyak digunakan pada bidang medis, contoh rekontruksi citra yaitu foto *rontgen* digunakan untuk membentuk ulang citra organ tubuh.

Di dalam mengolah sebuah citra, terdapat berbagai algoritma yang dapat diterapkan untuk menghasilkan keluaran yang lebih baik. Keluaran yang baik akan mempengaruhi hasil dari proses yang akan dilanjutkan selanjutnya.

1. Citra Digital

Citra (*image*) secara harfiah merupakan gambar pada bidang dwimatra (dua dimensi). Sedangkan ditinjau dari sudut pandang matematis, citra merupakan fungsi menerus (*continue*) dari intensitas cahaya pada bidang dwimatra. Sumber cahaya menerangi objek kemudian memantulkan kembali sebagian dari berkas cahaya tersebut. Pantulan cahaya ini ditangkap oleh alat-alat optik, misalnya mata pada

manusia, kamera, pemindaian (*scanner*) dan sebagainya, sehingga bayangan objek yang disebut citra tersebut terekam (Permadi & Murinto, 2015).

Berikut adalah elemen-elemen yang terdapat pada citra digital (Sutoyo, 2009:24):

1. Kecerahan (*Brightness*)

Brightness merupakan intensitas cahaya yang dipancarkan piksel dari citra yang dapat ditangkap oleh sistem penglihatan. Kecerahan pada sebuah titik (piksel) di dalam citra merupakan intensitas rata-rata dari suatu area yang melingkupinya.

2. Kontras (*Contrast*)

Kontras menyatakan sebaran terang dan gelap dalam sebuah citra. Pada citra yang baik, komposisi gelap dan terang tersebar secara merata.

3. Kontur (*Contour*)

Kontur adalah keadaan yang ditimbulkan oleh perubahan intensitas pada piksel-piksel yang bertetangga. Karena adanya perubahan intensitas inilah mata mampu mendeteksi tepi-tepi objek di dalam citra.

4. Warna

Warna sebagai persepsi yang ditangkap sistem visual terhadap panjang gelombang cahaya yang dipantulkan oleh objek.

5. Bentuk (*Shape*)

Shape adalah properti intrinsik dari objek 3 dimensi, dengan pengertian bahwa bentuk merupakan properti intrinsik utama untuk sistem visual manusia.

6. Tekstur (*Texture*)

Texture dicirikan sebagai distribusi spasial dari derajat keabuan di dalam sekumpulan piksel-piksel yang bertetangga. Tekstur adalah keteraturan pola-pola tertentu yang terbentuk dari susunan piksel-piksel dalam citra digital. Informasi tekstur dapat digunakan untuk membedakan sifat-sifat permukaan suatu benda dalam citra yang berhubungan dengan kasar dan halus, juga sifat-sifat spesifik dari kekasaran dan kehalusan permukaan tadi, yang sama sekali terlepas dari warna permukaan tersebut

2. Jenis citra

Nilai suatu *pixel* memiliki nilai dalam rentang tertentu, dari nilai minimum sampai nilai maksimum. Jangkauan yang digunakan berbedabeda tergantung dari jenis warnanya. Citra dengan penggambaran seperti ini digolongkan ke dalam citra integer.

Berikut adalah jenis-jenis citra berdasarkan nilai *pixel*-nya (Zaenury Ichsan et al., 2014), sebagai berikut:

a. Citra biner

Citra biner adalah citra digital yang hanya memiliki dua kemungkinan nilai *pixel* yaitu hitam dan putih. Citra biner juga disebut sebagai citra B&W (*black and white*) atau monokrom.

b. Citra *grayscale*

Citra *grayscale* memberi kemungkinan warna yang lebih banyak daripada citra biner, karena ada nilai-nilai lain diantara nilai minimum (biasanya = 0) dan nilai maksimumnya. Citra *grayscale* disebut juga citra keabuan karena pada umumnya warna yang dipakai adalah antara hitam sebagai warna minimal dan warna putih sebagai warna maksimal, sehingga warna antaranya adalah abu-abu. Ada beberapa macam untuk mengkonversi sistem warna RGB menjadi *grayscale* yaitu:

1. Dengan merata-rata setiap komponen warna RGB

$$Grayscale = \frac{R+G+B}{3} \quad (2.1)$$

Keterangan:

R = *Red* (merah)

G = *Green* (hijau)

B = *Blue* (biru)

2. Dengan nilai maximal dari nilai RGB $Grayscale = Max[R, B, G]$
3. Dengan menggunakan YUV (sistem pada NTSC) yaitu dengan cara mengambil komponen Y (iluminasi). Komponen Y dapat diperoleh dari sistem warna RGB dengan konversi: $Grayscale = 0.299R + 0.587G + 0.114B$.

c. Citra warna (16 bit)

Citra warna 16 bit (biasanya disebut sebagai citra *high colour*) dengan setiap *pixel* diwakili dengan 2 byte memory (16bit). Warna 16 bit memiliki warna 65536 warna. Dalam formasi bitnya, nilai merah dan biru mengambil tempat di 5 bit di kanan dan di kiri. Komponen hijau memiliki 5 bit ditambah 1 bit ekstra. Pemilihan komponen hijau dengan deret 6 bit dikarenakan penglihatan manusia lebih sensitif terhadap warna hijau.

2.3. Visi Komputer

Computer vision adalah proses otomatis yang mengintegrasikan sejumlah besar proses untuk persepsi visual, seperti akuisisi citra, pengolahan citra, pengenalan dan

pembuatan keputusan. *Computer vision* mencoba meniru cara kerja sistem visual manusia (*human vision*) yang sesungguhnya sangat kompleks, bagaimana manusia melihat objek dengan indra penglihatan (mata), lalu citra objek diteruskan ke otak untuk diinterpretasi sehingga manusia mengerti objek apa yang tampak dalam pandangan mata. Selanjutnya hasil interpretasi ini digunakan untuk pengambilan keputusan. (Darmawan, 2009).

Dalam melakukan pengenalan sebuah objek di antara banyak objek dalam citra, komputer harus melakukan proses segmentasi terlebih dahulu. Segmentasi adalah memisahkan citra menjadi bagian-bagian yang diharapkan merupakan objek-objek tersendiri atau membagi suatu citra menjadi wilayah-wilayah yang homogen berdasarkan kriteria keserupaan tertentu antara derajat keabuan suatu *pixel* dengan derajat keabuan *pixel-pixel* tetangganya. Menurut Darmawan (Darmawan, 2009).

Berdasarkan cara kerjanya, terdapat 2 jenis teknik segmentasi citra, yaitu:

- a. Segmentasi berdasarkan intensitas warna (derajat keabuan).

Berasumsi bahwa objek-objek yang akan dipisahkan cenderung memiliki intensitas warna yang berbeda-beda dan masing-masing objek memiliki warna yang hampir seragam. Salah satu teknik segmentasi berdasarkan intensitas warna adalah *mean clustering*. Pada *mean clustering* dilakukan pembagian citra dengan membagi histogram citra. Kelemahan segmentasi berdasarkan intensitas warna (derajat keabuan) antara lain adalah harus tahu dengan tepat berapa jumlah objek yang ada pada citra serta citra hasil kurang bagus jika pada citra terdapat beberapa objek dengan warna pada masing-masing objeknya bervariasi atau pada setiap objek memiliki warna yang sama.

- b. Segmentasi berdasarkan karakteristik.

Yaitu mengelompokkan bagian-bagian citra yang memiliki karakteristik yang sama berupa perubahan warna antara titik yang berdekatan, nilai rata-rata dari bagian citra tersebut. Untuk menghitung atau menentukan karakteristik digunakan perhitungan statistik, misalnya varian, *standard deviation*, teori probabilitas, transformasi *fourier*. Salah satu teknik segmentasi berdasarkan karakteristik adalah *split and merge*. Proses tersebut dilakukan secara rekursif karena pada setiap saat dilakukan proses yang sama tetapi dengan data yang selalu berubah (Wijaya & Prayudi, 2010).

Operasi-operasi pada pengolahan citra diterapkan bila (Prabowo & Abdullah, 2018):

1. Perbaikan atau memodifikasi citra dilakukan untuk meningkatkan kualitas penampakan citra/menonjolkan beberapa aspek informasi yang terkandung dalam

citra (*image enhancement*). Contoh: perbaikan kontras gelap/terang, perbaikan tepian objek, penajaman, pemberian warna semu, dll.

2. Adanya cacat pada citra sehingga perlu dihilangkan/ diminimumkan (*image restoration*). Contoh: penghilangan kesamaran (*debluring*) citra tampak kabur karena pengaturan fokus lensa tidak tepat/kamera goyang, penghilangan *noise*.
3. Elemen dalam citra perlu dikelompokkan, dicocokkan atau diukur (*image segmentation*). Operasi ini berkaitan erat dengan pengenalan pola.
4. Diperlukannya ekstraksi ciri-ciri tertentu yang dimiliki citra untuk membantu dalam pengidentifikasian objek (*image analysis*). Proses segmentasi kadangkala diperlukan untuk melokalisasi objek yang diinginkan dari sekelilingnya. Contoh: pendeteksian tepi objek.

2.4. Gaussian Filter

Mask yang sering pula digunakan untuk penghalusan citra adalah *mask* penghalusan *Gaussian* (*Gaussian smoothing*). Menurut Usman (2005:70), *Filter Gaussian* sangat baik untuk menghilangkan *noise* yang bersifat sebaran normal, yang banyak di jumpai pada sebaran citra hasil proses digitasi menggunakan kamera karena merupakan fenomena alamiah akibat sifat pantulan cahaya dan kepekaan sensor cahaya pada kamera itu sendiri. *Gaussian Blur* adalah *Filter blur* yang menempatkan warna transisi yang signifikan dalam sebuah *image*, kemudian membuat warna-warna pertengahan untuk menciptakan efek lembut pada sisi-sisi sebuah *image* (Sunandar, 2017).

Bobot pada *mask* penghalusan *Gaussian* mengikuti distribusi normal sebagaimana yang dinyatakan dalam persamaan di bawah ini:

$$h(m, n) = \frac{1}{2\pi\sigma^2} e^{-\frac{(m^2+n^2)}{2\sigma^2}} \quad (2.2)$$

Dimana :

1. σ adalah nilai *standard deviation* distribusi normal yang digunakan. Makin besar nilai σ , maka makin banyak titik tetangga yang diikutkan dalam perhitungan.
2. m dan n adalah posisi koordinat *mask* dimana koordinat (0,0) adalah posisi titik tengah dari *mask* yang mempunyai nilai paling besar/paling tinggi.
3. π adalah konstanta dengan nilai 3.14
4. e adalah konstanta bilangan natural dengan nilai 2, 718281828.

2.5. RGB Color Channel

Color channel menyimpan sebuah informasi warna dalam salah satu komponen warna utama. *Color channel* yang digunakan adalah RGB. Fitur warna yang digunakan menggunakan fitur *color moment* yang terdiri dari *mean* dan *standard deviation* pada masing-masing *channel* RGB (Hutagaol, Sari, & Adikara, 2019).

1. Mean

Menurut (Sari, Dewi, & Fatichah, 2014) *mean* adalah nilai rata-rata *pixel* yang akan dicari pada setiap *channel* RGB. Rumus untuk memperoleh fitur *mean* digunakan persamaan (2.3).

$$\mu = \frac{1}{M \times N} \sum_{i=1}^M \sum_{j=1}^N P_{ij} \quad (2.3)$$

Keterangan:

$$\mu = \text{Mean}$$

$$M \times N = \text{Ukuran Citra}$$

$$P = \text{Pixel Citra}$$

2. Standard deviation

Dasar perhitungan *standard deviation* adalah keinginan untuk mengetahui keragaman suatu kelompok data dengan mengurangi setiap nilai data dengan rata-rata kelompok data tersebut atau menunjukkan ukuran rata-rata kontras dari suatu citra. Untuk memperoleh fitur ini menurut (Dewi & Ginardi, 2014) digunakan rumus pada persamaan (2.4).

$$\sigma = \sqrt{\frac{1}{M \times N} \sum_{i=1}^M \sum_{j=1}^N (X_{ij} - \bar{X})^2} \quad (2.4)$$

Keterangan:

$$\sigma = \text{Standard deviation}$$

$$M \times N = \text{Ukuran Citra}$$

$$X = \text{Pixel Citra}$$

$$\bar{X} = \text{Mean}$$

2.6. Segmentasi Warna menggunakan *Hue, Saturation, dan Value* (HSV)

Segmentasi warna merupakan proses segmentasi dengan pendekatan daerah yang bekerja dengan menganalisis nilai warna dari tiap *pixel* pada citra dan membagi citra tersebut sesuai dengan fitur yang diinginkan. Segmentasi citra dengan deteksi warna HSV menggunakan dasar seleksi warna pada model warna HSV dengan nilai toleransi tertentu.

Pada metode segmentasi dengan deteksi warna HSV, dilakukan pemilihan sampel *pixel* sebagai acuan warna untuk membentuk segmen yang diinginkan (Yulian Fauzi, 2011).

Citra digital menggunakan model warna RGB sebagai standar acuan warna, oleh karena itu proses awal pada metode ini memerlukan konversi model warna RGB ke HSV. Ruang warna HSV terdiri dari 3 komponen, yaitu: H menunjukkan jenis warna (seperti merah, biru atau kuning) atau corak warna, yaitu tempat warna tersebut ditemukan dalam spektrum warna. S mewakili tingkat dominasi warna yaitu ukum seberapa besar kemurnian dari warna tersebut. Sedangkan V mewakili tingkat kecerahan yaitu ukuran seberapa besar kecerahan suatu warna atau seberapa besar besar cahaya data dari suatu warna (Ericks Rachmat Swedia dan Margi Cahyanti, 2010). Untuk membentuk segmen sesuai dengan warna yang diinginkan maka ditentukan nilai toleransi pada setiap dimensi warna HSV, kemudian nilai toleransi tersebut digunakan dalam perhitungan proses *adaptive threshold* (Zarwani, 2018). Hasil dari proses *threshold* tersebut akan membentuk segmen area dengan warna sesuai toleransi yang diinginkan. Secara manual ruang warna RGB dapat dikonversikan ke dalam ruang warna HSV dengan melakukan perhitung terhadap nilai-nilai RGB itu sendiri dengan menggunakan rumus seperti berikut:

$$\begin{aligned} H &= \tan(3(G-B) / (R-G)+(R-B)) \\ S &= 1 - \min(R,G,B) / V \\ V &= (R+G+B) / 3 \end{aligned} \quad (2.5)$$

Model warna HSV di atas, merupakan model yang diperkenalkan oleh A.R Smith, namun kelemahan dari model ini yaitu jika nilai S yang didapat adalah 0 ($S = 0$) maka nilai H tidak terdefiniskan. Sehingga dibuatlah cara kedua oleh Acharya dan Ray dengan menggunakan model seperti berikut :

$$\begin{aligned} r &= \frac{R}{R+G+B} & g &= \frac{G}{R+G+B} & b &= \frac{B}{R+G+B} \\ V &= \max(r,g,b) \\ S &= \begin{cases} 0, & \text{jika } V = 0 \\ 1 - \frac{\min(r,g,b)}{V}, & V > 0 \end{cases} \\ H &= \begin{cases} 0, & \text{jika } S = 0 \\ \begin{cases} \frac{60*(g-b)}{S*V}, & \text{jika } V = r \\ 60 * [2 + \frac{b-r}{S*V}], & \text{jika } V = g \\ 60 * [4 + \frac{r-g}{S*V}], & \text{jika } V = b \end{cases} \end{cases} \\ H &= H + 360 \text{ jika } H < 0 \end{aligned} \quad (2.6)$$

Keterangan :

$H = Hue$ (warna sebenarnya)

$S = Saturation$ (kemurnian warna)

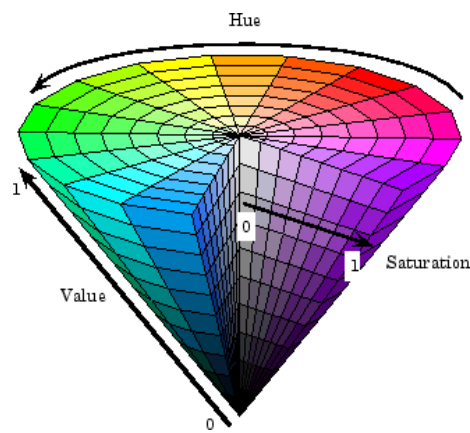
$V = Value$ (kecerahan)

$R/r = red$ (merah)

$G/g = green$ (hijau)

$B/b = blue$ (biru)

Gambar 2 menunjukkan ruang warna HSV sebagai berikut:



Gambar 2 Ruang Warna HSV

(Kadir dan Susanto, 2013)

2.7. Thresholding

Thresholding merupakan bagian dari teknik segmentasi yang banyak digunakan untuk membedakan antara latar belakang dan objek yang ada dengan mengkonversikan nilai intensitas ke dalam nilai 1 atau 0. *Thresholding* merupakan konversi citra berwarna ke citra *biner* yang dilakukan dengan cara mengelompokkan nilai derajat keabuan setiap pixel ke dalam 2 kelas, hitam dan putih. Pada citra hitam putih mempunyai nilai skala antara “0” sampai dengan “255” atau $[0,255]$, dalam hal ini nilai intensitas 0 menyatakan hitam, dan nilai intensitas 255 menyatakan putih, dan nilai antara 0 sampai 255 menyatakan warna keabuan yang terletak antara hitam dan putih. Pada operasi pengambangan, nilai intensitas *pixel* dipetakan ke salah satu dari dua nilai, α_1 atau α_2 (Zarwani, 2018). Rumus untuk menentukan nilai *threshold* bias didapatkan dari persamaan sebagai berikut:

$$g(x, y) = \begin{cases} 1 & \text{if } f(x, y) \geq T \\ 0 & \text{if } f(x, y) < T \end{cases} \quad (2.7)$$

Keterangan:

$g(x,y)$ = Nilai matriks citra hasil *thresholding*.

$f(x,y)$ = Merupakan nilai matriks citra yang akan di-*threshold*.

T = Merupakan nilai *threshold* (0 – 255).

2.8. Blob Detection

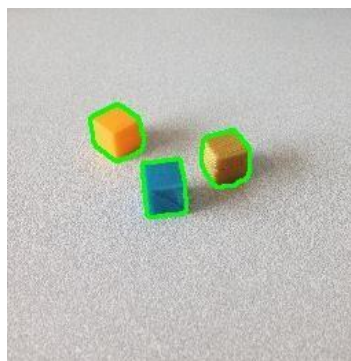
Dalam suatu *image processing* yang menggunakan segmentasi *foreground*, analisis *blob* merupakan teknik yang digunakan untuk menyatakan luas area *pixel* dari suatu *image* yang menjadi fokus deteksi untuk menentukan nilai *Blob*, ada beberapa hal yang harus diketahui untuk menghasilkan sebuah *blob* yang optimal. Penentuan luas *blob* pada setiap objek pada proses segmentasi *foreground* perlu dianalisis karena nilai *blob* pada tiap objek akan berbeda. Hal ini dipengaruhi oleh fitur objek seperti ukuran, jenis, dan teknik pengambilan data video (Rifaldi, 2017).

Prosesnya dimulai dari penandaan area *foreground* yang dianggap objek, kemudian pengumpulan data area menjadi *blob* seperti posisi *pixel* awal, panjang terhadap sumbu x dan sumbu y dan luas area *pixel* area sebuah *blob* (Rifaldi, 2017).

Metode *blob detection* terdiri dari langkah-langkah berikut:

1. Ekstraksi - proses yang diterapkan untuk mendapatkan wilayah objek tertentu yang sedang dicari. Proses ini biasanya dilakukan menggunakan segmentasi warna.
2. Perbaikan - wilayah yang diekstraksi diperbaiki dengan menghilangkan *noise* yang ada. Proses ini biasanya dilakukan menggunakan teknik transformasi wilayah.
3. Analisis - proses ini dilakukan untuk mengekstrak informasi yang dibutuhkan.

Berikut contoh area *blob* yang ditunjukkan pada **Error! Reference source not found.**

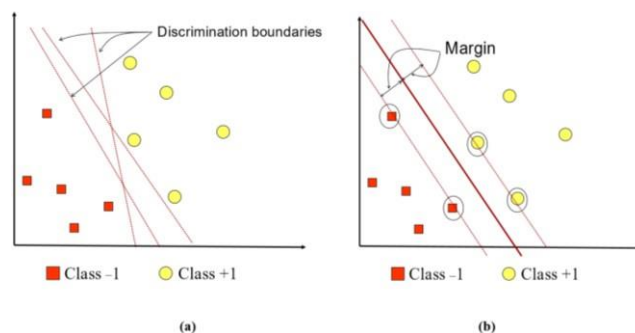


Gambar 3 Area Blob

(Sumber: layer0.authentise.com)

2.9. Support Vector Machine (SVM)

SVM merupakan salah satu metode terbaik yang bisa dipakai dalam permasalahan klasifikasi. Konsep SVM bermula dari masalah klasifikasi dua kelas sehingga membutuhkan training set positif dan negatif. SVM berusaha menemukan *hyperplane* (pemisah) terbaik untuk memisahkan ke dalam dua kelas dan memaksimalkan margin antara dua kelas tersebut. Prinsip dasar SVM adalah *linear classifier*, dan selanjutnya dikembangkan agar dapat bekerja pada problem *non-linear* dengan memasukkan konsep kernel *trick* pada ruang kerja berdimensi tinggi (Nurhikmah Arifin, 2019).



Gambar 4 Ilustrasi Pencarian Hyperplane Terbaik Menggunakan SVM

(Kesumawati, 2018)

Error! Reference source not found. memperlihatkan beberapa *pattern* yang merupakan anggota dari dua buah *class*: +1 dan -1. *Pattern* yang tergabung pada *class* -1 disimbolkan dengan warna merah (kotak), sedangkan *pattern* pada *class* +1, disimbolkan dengan warna kuning (lingkaran). Problem klasifikasi dapat diterjemahkan dengan usaha menemukan garis (*hyperplane*) yang memisahkan antara kedua kelompok tersebut. Berbagai alternatif garis pemisah (*discrimination boundaries*) ditunjukkan pada gambar 1-a. *Hyperplane* pemisah terbaik antara kedua *class* dapat ditemukan dengan mengukur *margin hyperplane* tersebut dan mencari titik maksimalnya. *Margin* adalah jarak antara *hyperplane* tersebut dengan *pattern* terdekat dari masing-masing *class*. *Pattern* yang paling dekat ini disebut sebagai *support vector*. Garis solid pada gambar 1-b menunjukkan *hyperplane* yang terbaik, yaitu yang terletak tepat pada tengah-tengah kedua *class*, sedangkan titik merah dan kuning yang berada dalam lingkaran hitam adalah *support vector*.

Pencarian lokasi *hyperplane* optimal merupakan inti dari metode SVM.

Diasumsikan bahwa terapat data learning dengan data points x_i ($i=1, 2, \dots, m$) memiliki

dua kelas $y_1 = \pm 1$ yaitu kelas positif (+1) dan kelas negatif (-1) sehingga akan diperoleh *decision function* berikut.

$$f(x) = \text{sign}(w \cdot x + b) \quad (2.8)$$

Dimana (\cdot) merupakan scalar sehingga $w \cdot x \equiv w \cdot x$

Berdasarkan pada decision di atas, dapat terlihat bahwa data akan terklasifikasi secara tepat jika $y_i (w \cdot x_i + b) > \forall i$ karena ketika $(w \cdot x_i + b)$ harus bernilai positif saat $y_i = +1$, dan bernilai negatif ketika $y_i = -1$. Decision function menjadi varian ketika akan dilakukan pembuatan skala positif baru dari argument dalam persamaan fungsi sehingga akan mengakibatkan ambiguitas dalam mendefinisikan skala untuk (w, b) dengan menetapkan $w \cdot x + b = 1$ untuk titik terdekat pada satu sisi dan $w \cdot x + b = -1$ untuk titik terdekat pada sisi lainnya. *Hyperplane* yang melewati $w \cdot x + b = 1$ dan $w \cdot x + b = -1$ disebut sebagai *hyperplane* kanonik dan wilayah antar *hyperplane* disebut sebagai *margin band*.

Margin maksimum dapat diperoleh dengan cara memaksimalkan nilai jarak antara *hyperplane* dan titik terdekatnya yaitu $\frac{1}{\|w\|}$. Hal tersebut dirumuskan sebagai *Quadratic*

Programming (QP) Problem dengan mencari titik minimal seperti pada persamaan berikut.

$$\min r(w) = \frac{1}{2} \|w\|^2 \quad (2.9)$$

Sedangkan subjek *constrain*/kendala persamaannya adalah sebagai berikut.

$$y_i (w \cdot x_i + b) \geq 1 \quad \forall i \quad (2.10)$$

Persamaan di atas merupakan permasalahan optimisasi kendala dimana kita meminimalkan fungsi objek pada persamaan (2.9) dengan kendala pada persamaan (2.10). Permasalahan di atas dapat direduksi dengan menggunakan fungsi *Lagrange* yang terdiri dari jumlahan fungsi objektif dan m kendala dikalikan dengan pengganda *Lagrange* seperti berikut.

$$L(w, b) = \frac{1}{2} (w \cdot w) - \sum_{i=1}^n \alpha_i (y_i (w \cdot x_i + b) - 1) \quad (2.11)$$

Dimana α_i merupakan *Lagrange Multipliers*, dan nilai $\alpha_i \geq 0$. Pada saat minimum, akan dilakukan penurunan dari b dan w dan mengaturnya menjadi nol seperti berikut.

$$\frac{\partial L}{\partial b} = - \sum_{i=1}^n \alpha_i y_i = 0 \quad (2.12)$$

$$\frac{\partial L}{\partial w} = - \sum_{i=1}^n \alpha_i y_i x_i = 0 \quad (2.13)$$

Substitusi nilai w dari persamaan (2.13) ke dalam bentuk $L(w, b)$ sehingga akan diperoleh rumus ganda atau biasa disebut sebagai *wolfe dual*.

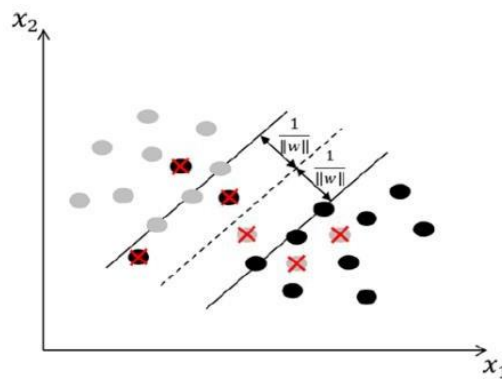
$$W(\alpha) = \sum_{i=1}^n \alpha_i - \sum_{i=1}^n \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) = 0 \quad (2.14)$$

Dimana nilai α_i terhadap kendala adalah sebagai berikut.

$$\alpha_i \geq 0 \quad \sum_{i=1}^m \alpha_i y_i = 0 \quad (2.15)$$

- *Soft – Margin SVM*

Ketika data yang digunakan tidak sepenuhnya dapat dipisahkan, *slack variables* x_i diperkenalkan ke dalam fungsi obyektif SVM untuk memungkinkan kesalahan dalam misklasifikasi. Dalam hal ini, SVM bukan lagi *hard margin classifier* yang akan mengklasifikasi semua data dengan sempurna melainkan sebaliknya yaitu SVM *soft margin classifier* dengan mengklasifikasikan sebagian besar data dengan benar, sementara memungkinkan model untuk membuat misklasifikasi beberapa titik di sekitar batas pemisah. Berikut merupakan gambar ketika data termasuk ke dalam *soft margin SVM* (Ningrum, 2018).



Gambar 5 Beberapa Misklasifikasi Pada Soft Margin SVM

(Buku Efficient Learning Machine, 2015)

Berdasarkan pada **Error! Reference source not found.** di atas, terlihat bahwa data pada kedua kelas tidak terpisah secara sempurna dapat dilihat dari beberapa lingkaran abu – abu yang persebarannya berada di sekitar area lingkaran hitam serta sebaliknya terdapat beberapa lingkaran hitam yang persebarannya berada di sekitar lingkaran abu – abu. Persamaan *soft margin* hampir mirip dengan *hard margin* hanya terdapat sedikit modifikasi dengan adanya *slack variable* pada persamaan (2.11) sebelumnya seperti berikut.

$$y_i (w \cdot x_i + b) \geq 1 - \varepsilon_i \quad (2.12)$$

Kemudian ketika akan dilakukan minimasi jumlahan eror $\sum_{i=1}^m \varepsilon_i$ adalah sebagai berikut.

$$\min \left[\frac{1}{2} w \cdot w + c \sum_{i=1}^m \alpha_i [y_i (w \cdot x_i + b) - 1 + \varepsilon_i] - \sum_{i=1}^m r_i \varepsilon_i \right] \quad (2.13)$$

Dengan demikian, persamaan (2.15) akan diubah ke dalam persamaan berikut.

$$\min r(w) = \frac{1}{2} \|w\|^2 + c \sum_{i=1}^m \varepsilon_i \quad (2.14)$$

Parameter C digunakan untuk mengontrol *trade off* antara *margin* dan kesalahan klasifikasi ε . C merupakan parameter yang menentukan besar kesalahan dalam klasifikasi

data dan nilainya ditentukan oleh pengguna. Peran C yaitu meminimalkan kesalahan pelatihan dan mengurangi kompleksitas model. Pemilihan parameter pada algoritma SVM dengan menggunakan metode *grid search* karena sangat handal jika diaplikasikan pada dataset yang mempunyai atribut sedikit daripada metode *random search* (Bergstra & Bengio, 2012).

SVM memiliki karakteristik sebagai berikut (Anto, 2003):

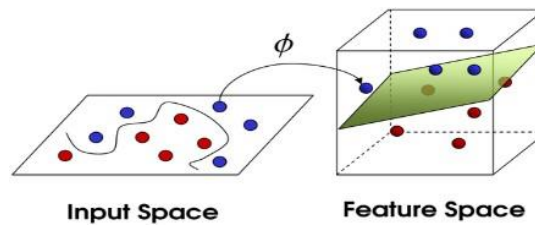
1. Secara prinsip SVM adalah *linear classifier*
2. *Pattern recognition* dilakukan dengan mentransformasikan data *input space* ke ruang yang berdimensi lebih tinggi, dan optimisasi dilakukan pada ruang vektor yang baru tersebut. Hal ini membedakan SVM dari solusi *pattern recognition* pada umumnya, yang melakukan optimisasi parameter pada ruang hasil transformasi yang berdimensi lebih rendah daripada dimensi *input space*.
3. Menerapkan strategi *Structural Risk Minimization* (SRM).
4. Prinsip kerja SVM pada dasarnya hanya mampu menangani klasifikasi dua kelas.

Pada dasarnya SVM adalah metode yang digunakan hanya untuk klasifikasi dua kelas (*binary classification*). Kemudian muncul beberapa metode diusulkan agar SVM mampu menyelesaikan permasalahan klasifikasi *multi-class* dengan cara mengombinasikan beberapa *binary classifier* (J.Z.Liang, 2004). Metode yang diusulkan adalah metode *one-against-one*. Adapun metode *one-against-one* ini akan dikonstruksi sejumlah $k(k-1)/2$ model klasifikasi SVM dengan masing-masing model yang ada dilatih menggunakan data dari dua kelas yang berbeda. Dengan demikian data pada kelas i dan j SVM akan menyelesaikan permasalahan klasifikasi *biner* untuk *multi-class*. Penelitian ini menggunakan metode *one-against-one*.

- Kernel SVM

Ketika terdapat permasalahan data yang tidak terpisah secara *linear* dalam ruang input, *soft margin* SVM tidak dapat menemukan *hyperplane* pemisah yang kuat yang meminimalkan misklasifikasi dari data points serta menggeneralisasi dengan baik. Untuk itu, kernel dapat digunakan untuk mentransformasi data ke ruang berdimensi lebih tinggi yang disebut sebagai ruang kernel, dimana akan menjadikan data terpisah secara *linear* (Ningrum, 2018). Hal ini sejalan dengan teori *Cover* yang menyatakan “*Jika suatu transformasi bersifat non linear dan dimensi dari feature space cukup tinggi, maka data pada input space dapat dipetakan ke feature space yang baru, dimana pattern-pattern tersebut pada probabilitas tinggi dapat dipisahkan secara linear*”.

Berikut **Error! Reference source not found.** yang menunjukkan kernel SVM untuk memisahkan data secara linear.



Gambar 6 Kernel SVM Untuk Memisahkan Data Secara Linear

(www.quora.com)

Data disimpan dalam bentuk kernel yang mengukur kesamaan atau ketidaksamaan objek data. Kernel dapat dibangun untuk berbagai objek data mulai dari data kontinu dan data diskrit melalui urutan data dan grafik. Konsep substitusi kernel berlaku bagi metode lain dalam analisis data. tetapi SVM merupakan yang paling terkenal dari metode dengan jangkauan kelas luas yang menggunakan kernel untuk merepresentasikan data dan dapat disebut sebagai metode berbasis kernel. Berikut merupakan ilustrasi contoh dalam melakukan pemisahan data menggunakan kernel. Diketahui bahwa data terdiri dari *input space* dengan dua buah $\mathbf{X} = \{x_1 \ x_2\}$ dan $\mathbf{Z} = \{z_1 \ z_2\}$. Diasumsikan fungsi kernel akan dibuat dengan menggunakan input x dan z seperti berikut.

$$\mathbf{K}(\mathbf{x}, \mathbf{z}) = (\mathbf{x}^T \mathbf{z})^2$$

$$\mathbf{K}(\mathbf{x}, \mathbf{z}) = (x_1 z_1 + x_2 z_2)^2$$

$$\mathbf{K}(\mathbf{x}, \mathbf{z}) = (x_1 z_1 + x_2 z_2 + 2x_1 z_2 x_1 z_2)^2$$

$$\mathbf{K}(\mathbf{x}, \mathbf{z}) = (x_1^2, \sqrt{2}x_1 x_2, x_2^2)^T + (z_1^2, \sqrt{2}z_1 z_2, z_2^2)$$

$$\mathbf{K}(\mathbf{x}, \mathbf{z}) = \Phi(\mathbf{x})^T \Phi(\mathbf{z}) \tag{2.15}$$

Nilai \mathbf{K} di atas secara implisit mendefinisikan pemetaan ke ruang dimensi yang lebih tinggi seperti berikut.

$$\Phi(\mathbf{x}) = \{ x_1^2, \sqrt{2}x_1 x_2, x_2^2 \} \tag{2.16}$$

Kernel $\mathbf{K}(\mathbf{x}, \mathbf{z})$ mengambil dua *input space* dan memberikan kesamaannya dalam *feature space* seperti berikut.

$$\Phi : X \rightarrow F$$

$$\mathbf{K} : X \times X \rightarrow R, \mathbf{K}(\mathbf{x}, \mathbf{z}) = \Phi(\mathbf{x}) \cdot \Phi(\mathbf{z})$$

Berdasarkan pada fungsi kernel di atas, dapat dilakukan perhitungan untuk melakukan prediksi dari beberapa data dalam *feature space* seperti pada persamaan berikut.

$$f(\Phi(x)) = \text{sign}(w \cdot \Phi(z) + b)$$

$$f(\Phi(x)) = \text{sign}\left(\sum_{i=1}^m a_i y_i \mathbf{K}(x, z) + b\right) \quad (2.17)$$

Dimana :

b : Nilai bias

m : Jumlah *support vector*

$\mathbf{K}(x, z)$: Fungsi nilai kernel

Nilai k yang bisa digunakan sebagai fungsi kernel harus memenuhi kondisi *Mercer* antara lain:

- Merupakan *Hilbert Space* dimana nilai *feature space* harus merupakan vektor dengan *dot product*.
- Harus benar jika k merupakan fungsi *definit* positif

$$\int dx \int dz f(x) \mathbf{K}(x, z) f(z) > 0 \quad (\forall f \in L_2) \quad (2.18)$$

- Ketika k_1 dan k_2 merupakan fungsi kernel, maka:

$$\mathbf{K}(x, z) = \mathbf{K}_1(x, z) + \mathbf{K}_2(x, z) \quad : \text{Direct sum} \quad (2.19)$$

$$\mathbf{K}(x, z) = a\mathbf{K}_1(x, z) \quad : \text{Skalar product} \quad (2.20)$$

$$\mathbf{K}(x, z) = \mathbf{K}_1(x, z)\mathbf{K}_2(x, z) : \text{Direct product} \quad (2.21)$$

Berikut merupakan fungsi kernel yang populer dan sering digunakan antara lain sebagai berikut.

1. *Linear* Kernel SVM

Linear kernel merupakan fungsi kernel yang paling sederhana. *Linear* kernel digunakan ketika data yang dianalisis sudah terpisah secara *linear*. *Linear* kernel cocok ketika terdapat banyak fitur dikarenakan pemetaan ke ruang dimensi yang lebih tinggi tidak benar – benar meningkatkan kinerja seperti pada klasifikasi teks. Dalam klasifikasi teks, baik jumlah *instances* (dokumen) maupun jumlah fitur (kata) sama sama besar. Berikut merupakan persamaan dari *linear* kernel SVM.

$$\mathbf{K}(x, z) = \mathbf{x}^T \mathbf{z} \quad (2.22)$$

Pemetaan fungsi Φ merupakan identitas/tidak ada pemetaan

2. *Polynomial* Kernel (derajat d)

Polinomial kernel merupakan fungsi kernel yang digunakan ketika data tidak terpisah secara *linear*. Polinomial kernel sangat cocok untuk permasalahan dimana semua training dataset dinormalisasi.

$$K(\mathbf{x}, \mathbf{z}) = (\mathbf{x}^T \mathbf{z})^d \text{ atau } (1 + \mathbf{x}^T \mathbf{z})^d \quad (2.23)$$

3. *Radial Basis Function* (RBF) Kernel

RBF kernel merupakan fungsi kernel yang biasa digunakan dalam analisis ketika data tidak terpisah secara *linear*. RBF kernel memiliki dua parameter yaitu *Gamma* dan *Cost*. Parameter *Cost* atau biasa disebut sebagai *C* merupakan parameter yang bekerja sebagai pengoptimalan SVM untuk menghindari misklasifikasi di setiap sampel dalam training dataset. Parameter *Gamma* menentukan seberapa jauh pengaruh dari satu sampel training dataset dengan nilai rendah berarti “jauh”, dan nilai tinggi berarti “dekat”. Dengan *gamma* yang rendah, titik yang berada jauh dari garis pemisah yang masuk akal dipertimbangkan dalam perhitungan untuk garis pemisah. Ketika *gamma* tinggi berarti titik-titik berada di sekitar garis yang masuk akal akan dipertimbangkan dalam perhitungan. Berikut merupakan persamaan dari RBF kernel.

$$K(\mathbf{x}, \mathbf{z}) = \exp[-\gamma \|\mathbf{x} - \mathbf{z}\|^2] \quad (2.24)$$

Pemilihan kernel sangat bergantung pada data spesifik. Sebagai contoh, Kernel *polynomial* banyak digunakan dalam pemrosesan gambar, sedangkan kernel *Gaussian* adalah kernel serba guna yang sebagian besar diterapkan tanpa adanya pengetahuan sebelumnya. Matriks kernel yang berakhir diagonal menunjukkan bahwa ruang fitur redundan dan kernel lain harus dicoba setelah pengurangan fitur. Perhatikan bahwa ketika kernel digunakan untuk mengubah vektor fitur dari ruang input ke ruang kernel untuk dataset yang tidak dapat dipisahkan secara *linear*, perhitungan matriks kernel membutuhkan memori yang besar dan sumber daya komputasi, untuk data besar (Ningrum, 2018).

2.10. Multi Kelas SVM

Saat pertama kali diperkenalkan oleh Vapnik, SVM hanya dapat mengklasifikasikan data ke dalam dua kelas. Namun, setelah dilakukan berbagai penelitian lebih lanjut untuk dapat mengembangkan SVM sehingga bisa mengklasifikasikan data yang memiliki lebih dari 2 kelas. Adapun 2 pilihan untuk mengimplementasikan *multi class* SVM yaitu pertama, dengan menggabungkan beberapa SVM biner dan yang kedua adalah menggabungkan semua data yang terdiri dari beberapa kelas ke dalam sebuah bentuk permasalahan optimasi. Namun pada pendekatan yang kedua permasalahan optimasi yang harus diselesaikan jauh lebih rumit (Banyal & Dayat, 2016). Berikut adalah implementasi *multi class* SVM dengan pendekatan pertama yang paling umum digunakan.

1. Metode *One against All*

Dengan menggunakan metode One-against-All, dibangun k buah model SVM biner (k adalah jumlah kelas). Setiap model klasifikasi ke-i dilatih dengan

menggunakan keseluruhan data, untuk mencari solusi permasalahan pada persamaan 2.25 (Pricila, 2016).

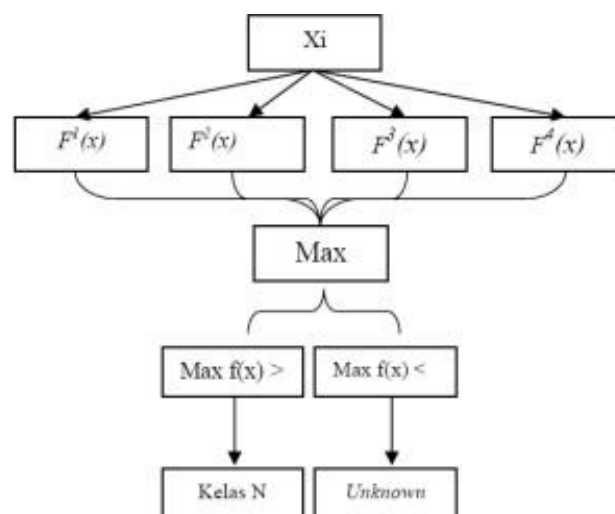
$$\begin{aligned}
 & \min_{w^i, b^i, \xi^i} \frac{1}{2} (w^i)^T w^i + C \sum_t \xi_t^i \\
 & s. t \ (w^i)^T \phi(x_t) + b \geq 1 - \xi_t^i \rightarrow y_t = i, \\
 & (w^i)^T \phi(x_t) + b \geq -1 + \xi_t^i \rightarrow y_t \neq i, \\
 & \xi_t^i \geq 0
 \end{aligned} \tag{2.25}$$

Contohnya, terdapat permasalahan klasifikasi dengan 4 buah kelas. Untuk pelatihan digunakan 4 buah SVM biner seperti pada **Error! Reference source not found.** (Pricila, 2016).

Tabel 1 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	$f^4(x) = (w^4)x + b^4$

Penggunaan dalam mengklasifikasi data baru dapat dilihat pada **Error! Reference source not found..**



Gambar 7 Contoh klasifikasi dengan metode One-Against-All (Permata, 2016)

Dalam menerapkan metode OAA akan dibangun z buah model SVM biner. z disini adalah jumlah kelas. Dalam mengklasifikasikan hal tersebut dapat dilihat pada persamaan:

$$\text{Kelas } g = \arg \max_{r=1,\dots,z} ((w^{(r)})^T \cdot \varphi [q_i] + b^{(r)}) \quad (2.26)$$

Dimana:

$(w^{(r)})^T = \text{hyperplane}$ yang jumlahnya sebanyak z

$\varphi [q_i] = \text{support vector}$

Hasil klasifikasi ditentukan dengan $\arg \max$ bahwa nilai tertinggi yang akan diambil dari hasil perhitungan semua *hyperplane* dengan *support vector* sebagai kelas targetnya atau juga biasa disebut sebagai kelas prediksi (Nugraha, 2019).

2. Metode *One against One*

Dengan menggunakan metode ini, dibangun $\frac{k(k-1)}{2}$ buah model klasifikasi biner (k adalah jumlah kelas). Setiap model klasifikasi dilatih pada data dari dua kelas. Untuk data pelatihan dari kelas ke- i dan kelas- j , dilakukan pencarian solusi untuk persoalan optimasi konstrain seperti pada persamaan 2.27 (Pricila, 2016).

$$\begin{aligned} \min_{w^{ij}, b^{ij}, \xi^{ij}} & \frac{1}{2} (w^{ij})^T w^{ij} + C \sum_t \xi_t^{ij} \\ \text{s. t. } & (w^{ij})^T \phi(x_t) + b^{ij} \geq 1 - \xi_t \rightarrow y_t = i, \\ & (w^{ij})^T \phi(x_t) + b^{ij} \geq -1 + \xi_t \rightarrow y_t = j, \\ & \xi_t^{ij} \geq 0 \end{aligned} \quad (2.27)$$

Jika data x dimasukkan ke dalam fungsi hasil pelatihan $f(x) = (w^{ij})^T \phi(x) + b$ 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 indeksinya lebih kecil dinyatakan sebagai kelas dari data. (Pricila, 2016). Contohnya, terdapat permasalahan klasifikasi dengan 4 buah kelas. Sehingga digunakan 6 buah SVM biner seperti pada Tabel 2.

Tabel 2 Contoh 6 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	$f^{34}(x) = (w^{34})x + b^{34}$

Berikut contoh penggunaannya dalam memprediksi kelas data baru dapat dilihat pada **Error! Reference source not found.**

Gambar 8 Contoh klasifikasi dengan metode One-Against-One (Pricila, 2016)