# SKRIPSI

# KLASIFIKASI MUTU CABAI MERAH BESAR (*Capsicuum Annuum* L.) BERBASIS VIDEO PROCESSING


**Disusun dan diajukan oleh**

**TUTI AMALIA**

**D421 16 014**




**DEPARTEMEN TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS HASANUDDIN**

**MAKASSAR**


**2022**

**LEMBAR PENGESAHAN SKRIPSI**

**KLASIFIKASI MUTU CABAI MERAH BESAR (Capsicuum Annuum L.) BERBASIS VIDEO PROCESSING**

**Disusun dan diajukan oleh**

**TUTI AMALIA**

**D42116014**

Telah dipertahankan di hadapan Panitia Ujian yang dibentuk dalam rangka Penyelesaian Studi Program Sarjana Program Studi Teknik Informatika Fakultas Teknik Universitas Hasanuddin pada tanggal 10 Juni 2022 dan dinyatakan telah memenuhi syarat kelulusan.

Menyetujui,

| Pembimbing Utama, | Pembimbing Pendamping, |
|---|---|
| Dr. Indrabayu, ST.,MT., M.Bus.Sys | Dr. Ir. Ingrid Nurtanio, M.T. |
| Nip. 197507162002121004 | Nip. 196108131988112001 |

Plt. Ketua Program Studi,

Dr. Amil Ahmad Ilham, S.T., M.IT.
Nip. 19731010 199802 1 002

## PERNYATAAN KEASLIAN

Yang bertanda tangan di bahwa ini :

| | |
|---|---|
| Name | : Tuti Amalia |
| NIM | : D42116014 |
| Program Studi | : Teknik Informatika |
| Jenjang | : S1 |

Menyatakan dengan ini bahwa Skripsi dengan judul

**KLASIFIKASI MUTU CABAI MERAH BESAR (*Capsicuum Annuum* L.) BERBASIS VIDEO PROCESSING**

adalah karya saya sendiri dan tidak melanggar hak cipta pihak lain. Apabila di kemudian hari Skripsi karya saya ini terbukti bahwa sebagian atau keseluruhannya adalah hasil karya orang lain yang saya pergunakan dengan cara melanggar hak cipta pihak lain, maka saya bersedia menerima sanksi.

Makassar, 25 Mei 2022

Yang menyatakan,


**TUTI AMALIA**

# KATA PENGANTAR

Alhamdulillah, puji syukur kehadirat Allah SWT atas berkat rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan tugas akhir ini yang berjudul "KLASIFIKASI MUTU CABAI MERAH BESAR (*Capsicuum Annuum* L.) BERBASIS VIDEO PROCESSING" sebagai salah satu persyaratan yang harus dipenuhi dalam menyelesaikan jenjang Strata-1 pada Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin. Sholawat serta salam kepada nabi Muhammad SAW yang telah menunjukkan dan mengajarkan akhlak mulia sehingga didapatkan kenyamanan dan keramahan dalam berhubungan dengan orang disekitar.

Dengan segala kerendahan hati, penulis menyadari bahwa dalam penyusunan dan penulisan laporan tugas akhir ini tidak lepas dari bantuan, bimbingan serta dukungan dari berbagai pihak, dari masa perkuliahan sampai dengan masa penyusunan tugas akhir ini. Sehingga, pada kesempatan ini penulis mengucapkan terimakasih yang sebesar-besarnya kepada:

1. Kedua Orang tua penulis, Bapak Burhanuddin dan Ibu Nasriyani yang selalu memberikan doa, dukungan semangat, dan motivasi, serta selalu sabar dalam mendidik penulis sejak kecil.
2. Bapak Dr. Indrabayu S.T., M.T., M.Bus.Sys. selaku pembimbing I, yang senantiasa memberikan saran-saran serta bantuan selama proses pengambilan data hingga selesainya sistem ini dibuat, dan Ibu Dr. Ir. Ingrid Nurtanio, M.T. selaku pembimbing II, yang senantiasa menyediakan waktu, tenaga, pikiran, semangat dan perhatian yang luar biasa dalam membimbing penulis menyusun tugas akhir ini.
3. Bapak Dr. Amil Ahmad Ilham, ST., M.I.T., selaku Ketua Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin yang telah membimbing penulis selama masa perkuliahan.
4. Segenap Dosen dan Staf Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin yang telah membantu dan memberikan banyak ilmu serta dukungan selama masa perkuliahan.

5. Sahabat-sahabat Ayu Lestari Ramadani, Irham Sahbana, Diki Siswanto, Raedi Radifan, Andi Amelia Ramadanti, Rizky Alfiansyah, Vickyarnoldo Wantura, Steve Lukis S.T, Ismayanti S.T, yang senantiasa membantu, menemani, dan memberikan semangat kepada penulis untuk menyelesaikan penyusunan tugas akhir ini.
6. Kakak-kakak, adik-adik AIMP *Research Group* FT UH, terutama teman-teman Cici Purnamasari, Putri Angriani, Nurul Musfirah, Asri Oktianawati, dan Dhinda Fitri Wiludjeng, Andi Marimar Muchtamar S.T, dan Riny Yustica Dewi S.T, yang telah memberikan begitu banyak bantuan selama penelitian, pengambilan data dan diskusi terkait progress penyusunan tugas akhir.
7. Saudara-saudara IGNITER16 FT-UH atas dukungan dan semangat yang diberikan selama ini kepada penulis.
8. Kak Nur Hikmah yang telah meminjamkan *conveyor belt* yang digunakan dalam pengambilan data penelitian.
9. Orang-orang berpengaruh lainnya yang tanpa sadar telah memotivasi dan membantu penulis.

Akhir kata dengan segala kerendahan hati, penulis menyadari tugas akhir ini masih banyak kekurangan, oleh karena itu kritik dan saran sangat diharapkan demi kesempurnaan tugas akhir ini. Penyusun berharap semoga tugas akhir ini dapat bermanfaat bagi semua pihak yang membacanya.

Makassar, Februari 2022

Penulis

# ABSTRAK

Penentuan mutu cabai selama ini dilakukan secara manual menggunakan pengamatan visual dengan memperhatikan bentuk fisik dan warna cabai tersebut. Pemutuan secara manual masih memiliki banyak kekurangan, diantaranya membutuhkan waktu yang relatif lama, menghasilkan produk sortasi yang beragam karena keterbatasan visual dan rabaan manusia, dan perbedaan persepsi (konsistensi) mutu produk hasil pemutuan karena unsur subyektifitas. Oleh karena itu, penelitian ini bertujuan untuk membuat sistem klasifikasi mutu cabai merah besar menggunakan metode SVM berbasis *video processing* yang dibagi menjadi empat kategori, yakni belum matang, setengah matang, matang, dan busuk. Dataset yang digunakan pada masing-masing kategori terdiri dari 35 cabai merah besar untuk tahap *training*. Pada tahap *testing* menggunakan video yang terdiri dari 40 cabai merah besar yang tercampur dan direkam secara acak untuk setiap kategori. Resolusi citra 1920 x 1080 piksel dan warna RGB. Metode yang digunakan untuk segmentasi yakni mengkonversi RGB ke HSV, *masking image* dengan menghasilkan citra biner yang digabungkan dengan hasil dari deteksi tepi menggunakan operator *Canny*, dan selanjutnya menggunakan operasi morfologi untuk pengikisan tepi objek. Klasifikasi mutu cabai merah besar menggunakan metode SVM. Parameter yang digunakan yakni kernel RBF, *Cost* (C) = 100 dan *gamma* ($\gamma$) = 0.2 yang diperoleh dari *Grid Search*, dan multi kelas OAA (*One Against All*). Pada penelitian ini, metode SVM dapat diimplementasikan dengan baik dengan hasil akurasi yang didapatkan sebesar 95.92%.

Kata kunci**:** Cabai merah besar, HSV, RBF, *Grid* Search, SVM, multi kelas OAA.

# DAFTAR ISI

# DAFTAR GAMBAR

# DAFTAR TABEL

# BAB I
# PENDAHULUAN

## 1.1 Latar Belakang

Implementasi Industri 4.0 di Indonesia khususnya sektor pertanian menjadi salah satu fokus pemerintah dalam meningkatkan daya saing dan produktivitas produk, terutama industri pengolahan makanan dan hasil pertanian (Ihsanuddin, 2018). Sektor pertanian sebagai bagian dari sistem agribisnis memiliki subsektor hortikultura dimana salah satu produk unggul dari subsektor hortikultura adalah tanaman sayuran. Cabai (*Capsicum annuum* L.) merupakan salah satu tanaman sayuran yang memiliki nilai ekonomis cukup tinggi (Istiqomah dkk., 2018). Secara umum cabai memiliki banyak kandungan gizi dan vitamin, diantaranya kalori, protein, lemak, karbohidrat, kalsium, vitamin A, B1, dan vitamin C (Fetri, 2019).

Cabai sebagai komoditas unggulan nasional memang mendapat perhatian serius dari pemerintah dan pelaku usaha karena daya saing dan adaptasinya yang menyumbang nilai ekonomi yang cukup tinggi terhadap perekonomian nasional. Berdasarkan data dari Badan Pusat Statistik dan Direktorat Jenderal Hortikultura, cabai termasuk ke dalam lima besar komoditas sayuran semusim dengan produksi mencapai 1,26 juta ton di tahun 2020 di mana produksi cabai setiap tahunnya cenderung meningkat dengan kenaikan sebesar 4,1% (49,77 ribu ton) dari tahun 2019. Sedangkan, produksi cabai di Sulawesi Selatan pada tahun 2020 mencapai 17 ribu ton (Badan Pusat Statistik, 2020).

Pada umumnya, cabai yang kadar airnya tinggi memiliki sifat mudah rusak (*perishable*). Sifat fisiologis ini menyebabkan cabai memiliki daya tahan dan umur simpan relatif singkat. Oleh karena itu, selain dikonsumsi segar cabai juga digunakan dalam produk olahan. Ada beberapa produk olahan cabai yang bisa memberikan nilai tambah, memperpanjang umur simpan dan mempertahankan mutu cabai, antara lain pengolahan cabai kering, cabai bubuk, abon cabai, saus atau sambal.

Permintaan masyarakat terhadap produk olahan ini cenderung meningkat setiap tahunnya. Terlebih, sebagian besar makanan cepat saji juga menggunakan

cabai olahan sebagai bahan pelengkap. Sebagai contoh, produsen mi cepat saji, seperti Indofood, Wings Food, dan Tiga Pilar, membutuhkan cabai untuk bumbu pelengkap mi sebanyak 5.000 ton per bulan. Perusahaan lain seperti PT Heinz ABC Indonesia juga memerlukan cabai sebagai salah satu bahan dasar utama pembuat saus. Berdasarkan data yang diperoleh, industri pengolahan cabai menyerap 10% dari pangsa pasar utama cabai (Hamid and Haryanto, 2012).

Untuk mendapatkan mutu produk yang memadai, pelaku usaha perlu memiliki dan menerapkan standar prosedur operasional untuk menghasilkan produk olahan yang memenuhi standar mutu yang dibutuhkan (Deptan, 2009). Cabai yang didatangkan dari petani tentunya membutuhkan pengolahan dan tahapan proses hingga menghasilkan sebuah produk olahan. Dalam skala industri kecil maupun menengah, kebersihan dan kualitas bahan baku merupakan kunci utama dari kualitas produk yang dihasilkan. Untuk menjaga kebersihan dan kualitas bahan baku dilakukan beberapa tahapan proses, diantaranya proses sortir cabai. Proses sortir atau sortasi adalah tindakan yang dilakukan untuk mendapatkan mutu yang baik dengan cara memilah-milah antara cabai yang baik dan tidak (Admin, 2020). Cabai yang baik adalah cabai yang matang, tidak muda, memiliki warna merah merata, segar, dan mulus (tidak cacat). Untuk ukuran cabai tidak menjadi kriteria utama dalam proses sortasi (Hamid and Haryanto, 2012). Penyortiran bahan menjadi penting karena penggunaan bahan tidak baik dapat mempengaruhi dan merusak kualitas produk olahan.

Banyak pelaku usaha belum memenuhi persyaratan kualitas dan keamanan pangan serta belum sesuai dengan tuntutan pasar yang terus berkembang, terutama pada pengolahan berskala rumah tangga, usaha kecil dan menengah. Penentuan mutu cabai selama ini dilakukan secara manual menggunakan pengamatan visual dengan memperhatikan bentuk fisik dan warna cabai tersebut (Khuriyati, 2020). Pemutuan secara manual masih memiliki banyak kekurangan, diantaranya membutuhkan waktu yang relatif lama, menghasilkan produk sortasi yang beragam karena keterbatasan visual dan rabaan manusia, dan perbedaan persepsi (konsistensi) mutu produk hasil pemutuan karena unsur subyektifitas (Sugianto, 2015). Pemilihan berdasarkan warna dari cabai merupakan salah satu faktor dalam

menentukan kualitas dan sebagai atribut sensori yang dapat diamati langsung sebagai indikator kesegaran dan kematangan, dimana terdapat perubahan warna dari warna hijau tua (cabai muda) menjadi merah (cabai matang).

Pada penelitian ini, metode yang digunakan untuk melihat perubahan warna cabai, yaitu metode citra digital atau *image processing*. Metode citra digital merupakan metode pengukuran indeks warna berbasis komputer yang diolah berdasarkan *file* citra yang diambil menggunakan kamera yang menghasilkan nilai *Red, Green,* dan *Blue.* Salah satu teknik pengolahan warna gambar adalah HSV (*Hue, Saturation, Value*). HSV merupakan salah satu sistem warna yang digunakan manusia dalam memilih warna objek. Nilai HSV diperoleh dari konversi warna RGB. Sistem HSV dipandang lebih dekat daripada sistem RGB dalam mendeskripsikan sensasi warna oleh mata manusia. Dengan menggunakan HSV, objek dengan warna tertentu dapat dideteksi dan mengurangi intensitas cahaya dari luar sehingga memudahkan dalam pemisahan *background* dan *foreground* (objek cabai).

Klasifikasi cabai pada sistem akan dibagi menjadi 4 kategori kelas yaitu cabai belum matang, setengah matang, matang, dan busuk. Algoritma yang digunakan dalam mengklasifikasi cabai yakni algoritma *Support Vector Machine* (SVM). Algoritma SVM memiliki keunggulan yaitu proses komputasi yang cepat dan bisa diimplementasikan dengan mudah (Octaviani dkk. 2014). Selain itu, SVM mampu menangani data yang bersifat linear dan non linear serta model klasifikasi pada SVM tetap baik meskipun dilatih dengan himpunan data yang sedikit (Suyanto, 2017).

## 1.2 Rumusan Masalah

Berdasarkan latar belakang, maka rumusan masalah yang akan diuraikan dalam skripsi ini antara lain:

a. Bagaimana sistem dapat mensegmentasi cabai merah besar?
b. Bagaimana hasil *testing* model dari sistem klasifikasi mutu cabai merah besar menggunakan metode SVM berbasis *video processing*?

## 1.3 Tujuan Penelitian

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

a. Membuat sistem yang dapat mensegmentasi cabai merah besar.
b. Mengetahui hasil *testing* model dari sistem klasifikasi mutu cabai merah besar menggunakan metode SVM berbasis *video processing*.

## 1.4 Manfaat Penelitian

Dengan dilakukannya penelitian ini, manfaat yang diharapkan dari penelitian ini adalah:

a. Membantu industri rumah tangga, kecil dan menengah dalam upaya peningkatan *quality control*.
b. Mendorong penggunaan teknologi *video processing* pada bidang industri, khususnya pada industri olahan dasar cabai merah besar.

## 1.5 Batasan Masalah Penelitian

Yang menjadi batasan masalah dari penelitian ini adalah sebagai berikut:

a. Objek penelitian berupa cabai merah besar yang dibagi menjadi 4 kategori berdasarkan mutu kematangan yaitu belum matang, setengah matang, matang, dan busuk.
b. Sistem klasifikasi mutu cabai merah besar menggunakan metode *Support Vector Machine* (SVM) berbasis *video processing*
c. Pengambilan data diambil dengan format video menggunakan *conveyor belt* dan kamera Logitech Webcam C922 Pro.
d. Data sampel yang digunakan untuk menentukan mutu cabai merah besar dari satu sisi yaitu bagian atas cabai.
e. Pengambilan data dilakukan dengan pencahayaan sebesar 15 lx dalam satuan Lux dengan alat ukur HT309.
f. Diasumsikan citra yang digunakan tidak tumpang tindih atau bertumpuk.

## 1.6 Sistematika Penulisan

Adapun sistematika penulisan pada penelitian ini adalah:

**BAB I PENDAHULUAN**

Bab ini berisi penjelasan tentang latar belakang, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah, dan sistematika penulisan.

**BAB II TINJAUAN PUSTAKA**

Bab ini berisi teori-teori terkait hal-hal yang mendasari dan yang berhubungan dengan penelitian, termasuk didalamnya cabai, visi komputer, pemrosesan citra, *machine learning*, dan metode yang digunakan.

**BAB III METODOLOGI PENELITIAN**

Bab ini berisi tentang apa saja yang akan dilakukan pada saat penelitian yang meliputi tahapan, waktu dan lokasi, instrumen penelitian, perancangan sistem, dan analisis kerja sistem.

**BAB IV HASIL DAN PEMBAHASAN**

Bab ini berisi tentang hasil penelitian dan pembahasan terkait pengolahan data yang telah dilakukan yang disertai dengan hasil penelitian.

**BAB V PENUTUP**

Bab ini akan dibahas mengenai kesimpulan yang diperoleh dari hasil penelitian yang dilakukan serta saran-saran untuk pengembangan sistem yang lebih lanjut.

# BAB II
# TINJAUAN PUSTAKA

## 2.1 Cabai

Menurut klasifikasi dalam tata nama (sistem tumbuhan) tanaman cabai termasuk ke dalam:

Divisi : *Spermatophyta*
Sub divisi : *Angiospermae*
Kelas : *Dicotyledoneae*
Ordo : *Solanales*
Famili : *Solanaceae*
Genus : *Capsicum*
Spesies : *Capsicum annum* L.

Tanaman cabai merah (*Capsicum annuum* L.) adalah tanaman perdu dengan rasa buah pedas yang disebabkan oleh kandungan *capsaicin*. Secara umum cabai memiliki banyak kandungan gizi dan vitamin, diantaranya kalori, protein, lemak, karbohidrat, kalsium, vitamin A, B1, dan vitamin C (Subagyono dkk, 2010).

Dikutip dari Nurfalach (2010), terdapat beberapa jenis tanaman cabai secara umum diantaranya:

1. Cabai besar (*Capsicum annuum* L.). Buah cabai besar berukuran panjang berkisar 6-10 cm, diameter 0,7-1,3 cm. Cabai besar di Indonesia dibagi menjadi dua kelompok yaitu cabai merah besar dan cabai merah keriting. Permukaan buah cabai merah besar halus dan mengkilap serta mempunyai rasa pedas. Sedangkan cabai merah keriting bentuknya lebih ramping dengan cita rasa sangat pedas. Cabai besar dapat tumbuh subur di dataran rendah sampai dataran tinggi. Cabai merah memiliki ciri-ciri, yakni bentuk buah besar, panjang dan meruncing, buah yang muda berwarna hijau, sedangkan buah yang tua berwarna merah, kulit buah agak tipis, serta banyak terdapat biji dan rasanya agak pedas.
2. Cabai kecil atau cabai rawit (*Capsicum frutescens*). Buah cabai rawit berukuran panjang berkisar 2-3,5 cm dengan diameter 0,4-0,7 cm. Cita rasa

cabai rawit biasanya sangat pedas, walaupun ada yang tidak pedas. Variasi warna cabai rawit dari kuning, oranye, dan merah.

3. Cabai hibrida, cabai hibrida dikelompokkan ke dalam buah cabai besar. Cabai hibrida lebih tahan terhadap hama dan penyakit.
4. Cabai hias, bentuk buah menarik dan berwarna-warni namun tidak dapat dikonsumsi. Cabai ini digunakan sebagai penghias halaman.

Menurut (Hendrawan, 2021), tingkat kematangan cabai merah besar dapat dilihat dari perubahan warna, dan warna akan mempengaruhi kualitas atau mutu. Secara umum, cabai merah besar memiliki sembilan indeks kematangan, di mana perbedaan setiap indeks dapat dilihat berdasarkan perubahan warna yang semakin matang. Adapun indeks kematangan cabai besar dapat dilihat pada Tabel 2.1.

**Tabel 2. 1** Indeks Kematangan Cabai Besar

| Indeks ke- | Ciri-ciri |
|---|---|
| 1 | Hijau kekuningan muda |
| 2 | Hijau cerah dan mengkilat |
| 3 | hijau cerah, mengkilat, dan licin |
| 4 | hijau tua, berkilau, dan licin |
| 5 | hijau tua atau hijau kehitaman dan mengkilat |
| 6 | hijau bercampur merah, mengkilat, dan licin |
| 7 | dominan merah bercampur hijau, agak mengkilat, sangat licin |
| 8 | dominan berwarna merah dan ada sedikit hijau atau warna oranye-kemerahan secara keseluruhan, mengkilat, dan licin |
| 9 | keseluruhan berwarna merah tua, sedikit mengkilat, dan licin |

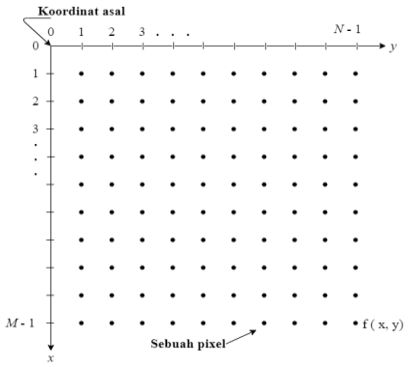
Dari sembilan karakteristik di atas kemudian dikelompokkan ke dalam 3 kategori yakni belum matang, setengah matang, dan matang. Selain itu terdapat penambahan kategori yakni cabai besar yang busuk. Untuk kategori belum matang diambil berdasarkan indeks ke-1 sampai ke-5, untuk setengah matang berdasarkan indeks ke-6 sampai ke-8, dan matang berdasarkan indeks ke-9. Sedangkan, untuk

kategori busuk berdasarkan studi literatur memiliki ciri warna yakni merah tua dan terdapat bercak hitam.

## 2.2 Pengolahan Citra Digital

Secara umum, pengolahan citra digital menunjuk pada pemrosesan gambar dua dimensi menggunakan komputer. Dalam konteks yang lebih luas, pengolahan citra digital mengacu pada pemrosesan setiap data dua dimensi. Citra digital merupakan sebuah larik (*array*) yang berisi nilai-nilai *real* maupun kompleks yang direpresentasikan dengan deretan bit tertentu (Putra D, 2010).

Menurut (Putra D, 2010), suatu citra dapat didefinisikan sebagai fungsi f(x, y) berukuran M baris dan N kolom, dengan x dan y adalah koordinat spasial, dan amplitudo f di titik koordinat (x, y) dinamakan intensitas atau tingkat keabuan dari citra pada titik tersebut. Apabila nilai x, y, dan nilai amplitudo f secara keseluruhan berhingga (*finite*) dan bernilai diskrit maka dapat dikatakan bahwa citra tersebut adalah citra digital. Gambar 2.1 menunjukkan posisi koordinat citra digital.




**Gambar 2. 1** Koordinat Citra Digital (Putra D, 2010)

Nilai pada suatu irisan antara baris dan kolom (pada posisi x, y) disebut dengan *picture elements*, *image elements*, *pels*, atau *pixels*. Istilah terakhir (piksel) paling sering digunakan pada citra digital.

Citra digital dapat ditulis dalam bentuk matriks sebagai berikut:

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,N-1) \\ f(1,0) & f(1,1) & \dots & f(1,N-1) \\ \vdots & \vdots & \ddots & \vdots \\ f(M-1,0) & f(M-1,1) & \dots & f(M-1,N-1) \end{bmatrix}$$

Pengolahan citra digital adalah manipulasi dan interpretasi digital dari citra dengan bantuan komputer. Pengolahan citra bertujuan untuk:

- Memperbaiki kualitas gambar, dilihat dari aspek *radiometric* dan aspek *geometric*. Aspek *radiometric* terdiri dari peningkatan kontras, restorasi citra, transformasi warna sedangkan aspek *geometric* terdiri dari rotasi, skala, translasi, transformasi *geometric.*
- Melakukan proses penarikan informasi atau deskripsi objek atau pengenalan objek yang terkandung pada citra.
- Melakukan pemulihan citra ciri (*feature images*) yang optimal untuk tujuan analisis.
- Melakukan kompresi atau reduksi data untuk tujuan penyimpanan data, transmisi data, dan waktu proses data.

Dalam melakukan pengolahan citra ada beberapa metodologi yang perlu dilakukan untuk mencapai sebuah tujuan. Adapun langkah-langkah sebagai berikut, (Ikhsanuddin, 2014):

1. Pembentukan Citra (*Data Acquisition*)

   Menentukan data yang diperoleh dan memilih metode perekam citra digital. Akuisisi citra merupakan proses menangkap (*capture*) atau memindai (*scan*) suatu citra analog sehingga diperoleh citra digital. Beberapa faktor yang perlu diperhatikan dalam proses akuisisi citra antara lain adalah: jenis alat akuisisi, resolusi kamera, teknik pencahayaan, perbesaran atau zooming, jarak, dan sudut pengambilan citra.
2. Pengolahan citra tingkat awal (*Image Processing*)

   Pada tahap ini dimana dalam meningkatkan kualitas citra dapat meningkatkan kualitas kontras, *brightness*, menghilangkan *noise*, perbaikan citra, transformasi, menghilangkan gangguan geometrik atau radiometrik, menentukan bagian citra yang akan diobservasi.

   Menurut (Sunandar, 2017), kecerahan (*brightness)* merupakan cahaya yang dipancarkan piksel dari citra yang dapat ditangkap oleh sistem penglihatan. Kecerahan pada sebuah titik (piksel) dalam citra yang merupakan intensitas rata-rata dari suatu area yang melingkupinya. Serta, kontras (*contrast*)

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

3. Segmentasi citra (*Image Segmentation*) dan deteksi sisi (*Edge Detection*)

   Proses dimana melakukan partisi citra menjadi wilayah-wilayah objek (*internal properties*) atau menentukan garis batas wilayah objek (*external shape characteristics*). Tahap ini bertujuan untuk mempartisi citra menjadi bagian-bagian pokok yang mengandung informasi penting misalnya memisahkan objek dan latar belakang segmentasi terdiri dari *downsampling*, penapisan dan deteksi tepian. Tahap *downsampling* merupakan proses untuk menurunkan jumlah piksel dan menghilangkan sebagian informasi dari citra. Dengan resolusi citra yang tetap, *downsampling* menghasilkan ukuran citra yang lebih kecil. Tahap segmentasi selanjutnya adalah deteksi tepian. Pendekatan tepi ini dirancang untuk mempresentasikan sebuah tepian yang ideal, dengan ketebalan yang diinginkan. Secara umum, proses segmentasi sangat penting dan secara langsung akan menentukan keakuratan sistem.

4. Seleksi dan Ekstraksi Ciri (*Feature Extraction and Selection*)

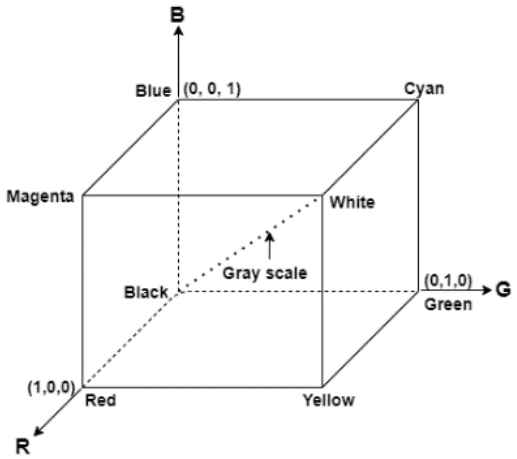
   Seleksi ciri dimaksudkan untuk memilih informasi kuantitatif dari ciri yang ada, yang dapat membedakan kelas-kelas objek secara baik. Ekstraksi ciri digunakan untuk mengukur besaran kuantitatif ciri di setiap piksel dari sebuah citra.

### 2.2.1. Konversi Tipe Citra

1. Citra warna
   - RGB

     Citra warna RGB merupakan jenis citra yang menyajikan warna dalam bentuk komponen R (merah), G (hijau), dan B (biru). Setiap komponen warna menggunakan 8 bit (nilainya berkisar antara 0 sampai dengan 255). Dengan demikian, kemungkinan warna yang bisa disajikan mencapai 255 x 255 x 255 atau 16.581.375 warna. Campuran ketiga warna primer tersebut dengan proporsi seimbang akan menghasilkan nuansa warna kelabu. Jika ketiga warna ini disaturasikan penuh, maka akan

menghasilkan warna putih (Pratt, 2007). Pada Gambar 2.2 menunjukkan pemetaan warna dalam ruang tiga dimensi.

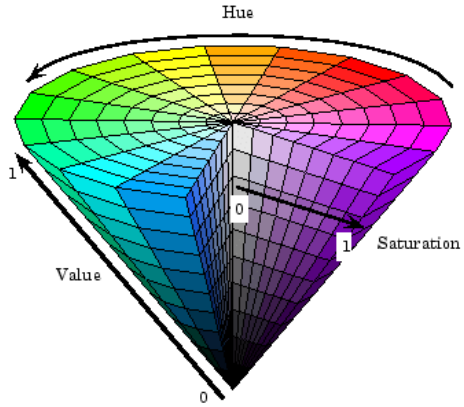



**Gambar 2. 2** Pemetaan warna dalam ruang tiga dimensi (Prianggodo, 2016)

- HSV

HSV (*Hue, Saturation, Value*) menunjukkan ruang warna dalam bentuk tiga komponen utama yaitu *hue*, *saturation*, dan *value* (atau disebut juga *brightness*). *Hue* adalah sudut dari 0 sampai 360 derajat. Biasanya 0 adalah merah, 60 derajat adalah kuning, 120 derajat adalah hijau, 180 derajat adalah cyan, 240 derajat adalah biru, dan 300 derajat adalah magenta seperti pada Gambar 2.3. *Hue* menunjukkan jenis warna (seperti merah, biru, kuning), yaitu tempat warna tersebut ditemukan dalam spektrum warna. Saturasi (*saturation*) suatu warna adalah ukuran seberapa besar kemurnian dari warna tersebut akibat pengaruh dari warna putih. Seperti warna merah, dengan pengaruh warna putih, warna merah menjadi bervariasi dari warna merah menuju merah muda, yang artinya *hue* masih tetap bernilai merah tetapi nilai saturasinya berkurang. Komponen HSV berikutnya adalah nilai value atau disebut intensitas, yaitu ukuran seberapa besar kecerahan suatu warna atau seberapa besar cahaya datang dari suatu warna. *Value* memiliki nilai dengan jangkauan 0% sampai 100% apabila nilainya 0 maka warnanya akan menjadi hitam (Pratt, 2007). Untuk merubah nilai RGB menjadi nilai HSV dapat

menggunakan teori travis, dapat dilihat pada persamaan 2.1 hingga 2.9 (Prianggodo, 2016).




**Gambar 2. 3** Model ruang warna HSV (Prianggodo, 2016)

Secara manual ruang warna RGB dapat dikonversikan ke dalam ruang warna HSV dengan melakukan perhitungan terhadap nilai-nilai RGB itu sendiri dengan menggunakan rumus sebagai berikut.

$$R' = \frac{R}{255} \tag{2.1}$$

$$G' = \frac{G}{255} \tag{2.2}$$

$$B' = \frac{B}{255} \tag{2.3}$$

$$Cmax = \max(R'.B'.C') \tag{2.4}$$

$$Cmin = \min(R'.B'.C') \tag{2.5}$$

$$\Delta = \text{Cmax} - \text{Cmin} \tag{2.6}$$

$$H = \begin{cases} 0 & ,\Delta = 0 \\ 60x\left(\frac{G'-B'}{\Delta} mod 6\right).Cmax = R' \\ 60x\left(\frac{B'-R'}{\Delta} + 2\right).Cmax = G' \\ 60x\left(\frac{R'-G'}{\Delta} + 4\right).Cmax = B' \end{cases} \tag{2.7}$$

$$S = \begin{cases} 0 & ,Cmax \\ \frac{\Delta}{Cmax} & ,Cmax \end{cases} \tag{2.8}$$

$$V = \text{Cmax} \tag{2.9}$$

Keterangan:

R = *red* (merah)

G = *green* (hijau)

B = *blue* (biru)

H = *Hue*

S = *Saturation*

V = *Value*

*2. Grayscaling*

*Grayscaling* adalah proses mengkonversi citra berwarna ke dalam bentuk citra *grayscale*, yaitu citra digital yang hanya memiliki satu nilai kanal pada setiap pikselnya, dengan kata lain nilai bagian *Red = Green = Blue*. Nilai tersebut digunakan untuk menunjukkan tingkat intensitas. Warna yang dimiliki adalah warna hitam, keabuan, dan putih. Tingkatan keabuan disini merupakan warna abu dengan berbagai tingkatan dari hitam hingga mendekati putih (Kanan, 2012). Formula yang digunakan untuk *grayscaling* adalah:

$$RGB\ (grayscale) = \frac{(R+G+B)}{3} = 0.333R + 0.333G + 0.333B \quad (2.10)$$

3. *Thresholding*

*Thresholding* atau binerisasi merupakan proses mengkonversi citra *grayscale* ke dalam bentuk citra biner. Citra biner sesuai namanya adalah citra yang tersusun atas dua nilai yaitu 0 atau 1. Jika piksel citra nilainya di atas nilai *threshold* maka piksel tersebut akan diubah ke warna putih dan nilainya menjadi 1. Sebaliknya jika nilai piksel citra berada di bawah *threshold* maka citra akan diubah ke warna hitam dan nilainya menjadi 0 (Prasetyo, 2011).

Salah satu cara untuk mengekstrak objek dari *background* adalah dengan memilih *threshold T* yang membagi mode-mode ini. Kemudian sembarang titik (x, y) untuk dimana f(x, y) ≥ disebut *object point*. Sedangkan yang lain disebut *background point*. Dengan kata lain, citra yang di-*threshold* g(x, y) didefinisikan sebagai berikut (Prasetyo, 2011).

$$g(x,y) = \begin{cases} 1 \; if \; f(x,y) \geq T \\ 0 \; if \; f(x,y) \leq T \end{cases} \quad (2.11)$$

Dimana, piksel yang diberi 1 berkaitan dengan objek sedangkan piksel yang diberi nilai 0 berkaitan dengan *background*.

### 2.2.2. Ekstraksi Fitur

Menurut (Sari dkk, 2017), ekstraksi fitur merupakan suatu pengambilan ciri atau *feature* dari suatu bentuk yang nantinya nilai yang didapatkan akan dianalisis untuk proses selanjutnya. Ekstraksi fitur dilakukan dengan cara menghitung jumlah titik atau piksel yang ditemui dalam setiap pengecekan. Pengecekan dilakukan dalam berbagai arah koordinat kartesian dari citra digital yang dianalisis, yaitu vertikal, horizontal, diagonal kanan, dan diagonal kiri. Fitur yang didapat dari sebuah citra merupakan ciri khas pembeda dengan citra-citra yang lain. Pada ekstraksi fitur warna, gambar dihitung menggunakan model warna RGB. Masing-masing *channel* warna R, G, dan B kemudian dihitung nilai *mean* atau rata-rata.

- Mean

  M*ean* adalah nilai rata-rata piksel yang akan dicari pada setiap *channel* RGB. Rumus untuk memperoleh fitur *mean* digunakan persamaan (2.12).

$$\mu = \frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} P_{ij} \quad (2.12)$$

Keterangan:

μ = *mean*

M = panjang gambar

N = lebar gambar

*P* = piksel citra

### *2.2.3. Blur*

*Blur* digunakan untuk mengaburkan citra dengan tujuan menghaluskan citra. Cara kerja *blur* adalah dengan menganalisis setiap piksel dalam sebuah gambar dan memadukannya dengan piksel tetangga untuk mengaburkan gambar.

Salah satu jenis teknik *blur* adalah *Gaussian blur*. *Gaussian blur* menggunakan kernel sendiri yaitu kernel *Gaussian*. *Gaussian blur* menyebabkan suatu citra menjadi kabur sehingga sudut-sudut tajam pada citra akan menjadi lebih halus. Pengolahan citra ini dapat berdampak suatu citra menjadi semakin baik, tapi bisa juga menjadi semakin buruk. Hasil citra yang telah mengalami *Gaussian blur* dapat dilihat pada Gambar 2.4, dimana hasil yang diberikan citra terlihat lebih halus dari citra aslinya (Gazali dkk, 2012).


**Gambar 2. 4** Blurring (a) Citra asli; (b) Citra yang telah mengalami Gaussian Blur (Gazali dkk, 2012)

Bobot pada *mask* penghalusan *Gaussian* mengikuti distribusi normal sebagaimana yang dinyatakan dalam persamaan 2.13.

$$G(i,j) = \frac{1}{2\pi\sigma^2} e^{\frac{(i-u)^2+(j-v)^2}{2\sigma^2}} \quad (2.13)$$

Keterangan:

$\sigma$ = standar deviasi (sigma)

$\pi$ = 3,14 (pi)

e = 2.71 (konstanta euler)

### 2.2.4. Morfologi Citra

Morfologi di dunia digital dapat diartikan sebuah cara untuk mendeskripsikan ataupun menganalisa bentuk dari objek digital. Morfologi dalam citra digital adalah suatu *tool* untuk ekstraksi komponen *image* yang berguna dalam representasi dan deskripsi dari bentuk daerah (*region shape*) dengan

*structuring element* (SE) untuk menentukan *properties of interest* dari *image* (Putra D, 2010).

Pada morfologi, suatu citra dinyatakan dengan himpunan koordinat diskrit (kontinu). Dalam hal ini, himpunan tersebut berhubungan dengan *point* atau piksel objek pada citra.

Karena objek dianggap sebagai suatu himpunan maka operasi-operasi himpunan seperti gabungan (*union*), irisan (*intersection*), dan komplemen (*complement*) dapat dilakukan (Putra D, 2010).

Operasi morfologi menggunakan dua input himpunan yaitu suatu citra (pada umumnya citra biner) dan suatu kernel. Khusus dalam morfologi, istilah kernel biasanya disebut *structuring elements* (elemen pembentuk struktur). SE merupakan suatu matriks yang pada umumnya berukuran kecil. Elemen dari SE dapat bernilai 1, 0, dan *don't care*. Nilai *don't care* biasanya ditandai dengan nilai elemen dikosongkan atau diberi tanda silang. Terdapat dua operasi dasar morfologi yaitu dilasi (*dilation*) dan erosi (*erosion*). Operasi-operasi ini menjadi dasar untuk membuat berbagai operasi morfologi yang sangat berguna untuk pengolahan citra digital, seperti *opening, closing, hit and miss transform, thinning,* dan *thickening* (Putra D, 2010).

Jika suatu objek (citra *input*) dinyatakan dengan A dan SE dinyatakan dengan B serta Bx menyatakan translasi B sedemikian sehingga pusat B terletak pada x. Operasi dilasi A dan B dapat dinyatakan sebagai berikut (Putra D, 2010).

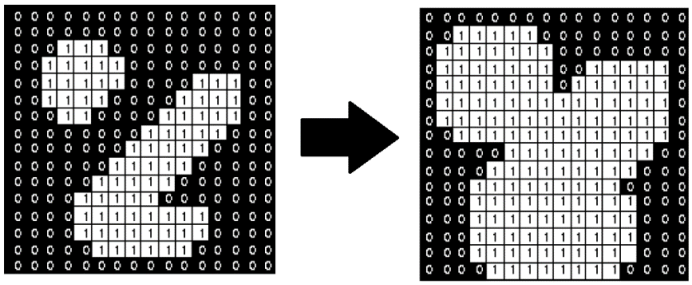
$$D(A,B) = A \oplus B = \{x : B_x \cap A \neq \emptyset\} \qquad (2.13)$$

Dengan $\emptyset$ menyatakan himpunan kosong.

Dilasi (*dilation*) berguna untuk memperluas atau menebalkan objek pada *image* biner. Proses dilasi dilakukan dengan membandingkan setiap piksel citra *input* dengan nilai pusat SE dengan cara melapiskan (*superimpose*) SE dengan citra sehingga pusat SE tepat dengan posisi piksel citra yang diproses. Jika paling sedikit ada 1 piksel pada SE sama dengan nilai piksel objek (*foreground*) citra maka piksel *input* diatur nilainya dengan nilai piksel *foreground* dan bila semua

piksel berhubungan adalah *background* maka *input* piksel diberi nilai piksel *background* (Putra D, 2010).

Semakin besar ukuran SE maka semakin besar perubahan yang terjadi. SE berukuran kecil juga dapat memberikan hasil yang sama dengan SE berukuran besar dengan cara melakukan dilasi berulang kali. Efek dilasi terhadap citra biner adalah memperbesar batas dari objek yang ada sehingga objek terlihat semakin besar dan lubang-lubang yang terdapat di tengah objek akan tampak mengecil (Putra D, 2010). Proses dilasi ditunjukkan pada Gambar 2.5.



**Gambar 2. 5** Proses dilasi dengan SE berukuran 3 x 3 dengan semua elemen SE bernilai 1 (Putra D, 2010)

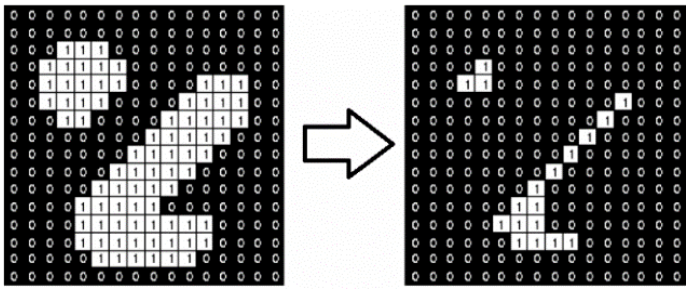
Secara matematis, operasi erosi dapat dinyatakan sebagai berikut.

$$E(A,B) = A \ominus B = \{x : B_x \subset X\} \tag{2.14}$$

Sama seperti dilasi, proses erosi dilakukan dengan membandingkan setiap piksel citra *input* dengan nilai pusat SE dengan cara melapiskan SE dengan citra sehingga pusat SE tepat dengan posisi piksel citra yang diproses. Jika semua piksel pada SE tepat sama dengan semua nilai piksel objek (*foreground*) citra maka piksel *input* diatur nilainya dengan nilai piksel *foreground*, jika tidak maka *input* piksel diberi nilai piksel *background*. Proses serupa dilanjutkan dengan menggerakkan SE piksel demi piksel pada citra *input* (Putra D, 2010).

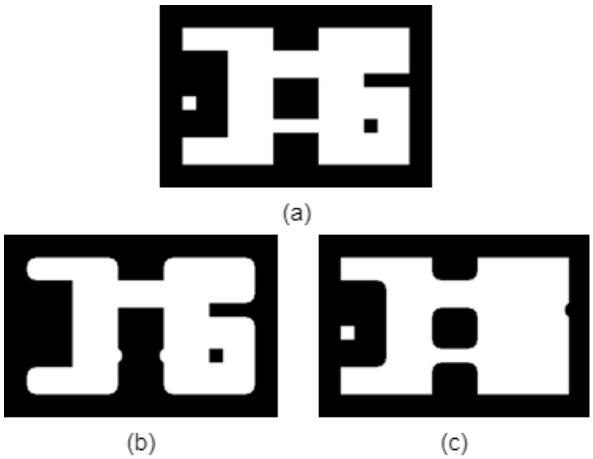
Proses erosi merupakan kebalikan dari proses dilasi. Jika dalam proses dilasi menghasilkan objek yang lebih luas maka dalam proses erosi akan menghasilkan objek yang menyempit (mengecil). Lubang pada objek juga akan membesar seiring menyempitnya objek tersebut (Putra D, 2010).

Dari Gambar 2.6 terlihat hasil proses erosi menyebabkan objek mengecil. Semakin besar kernel yang digunakan maka hasil yang akan didapatkan akan semakin kecil. Begitu juga apabila proses erosi dilakukan berulang-ulang akan terus mengecilkan objek walaupun hanya menggunakan SE berukuran kecil (Putra D, 2010).



**Gambar 2. 6** Proses erosi dengan SE berukuran 3 x 3 dengan semua elemen SE bernilai 1 (Putra D, 2010)

Kombinasi dari operasi erosi dan dilasi dapat membentuk operasi yang berbeda. Beberapa diantaranya yaitu *opening* dan *closing.* Dalam *opening*, pertama-tama dilakukan operasi erosi lalu diikuti dengan operasi dilasi. *Opening* seringkali digunakan untuk memisahkan piksel objek yang terlihat menonjol pada citra yang biasanya telah melewati proses *threshold* sebelumnya. Sedangkan pada *closing*, dilakukan operasi dilasi terlebih dahulu dan diikuti dengan erosi. *Closing* digunakan untuk menghubungkan komponen piksel objek pada citra. Hal tersebut dilakukan untuk mengurangi segmen yang tidak diinginkan. Proses morfologi citra pada operasi *opening* dan *closing* ditunjukkan pada Gambar 2.7.




**Gambar 2. 7** Morfologi citra (a) citra inputan; (b) hasil operasi *opening*; (c) hasil operasi *closing* (Susanto, 2019)

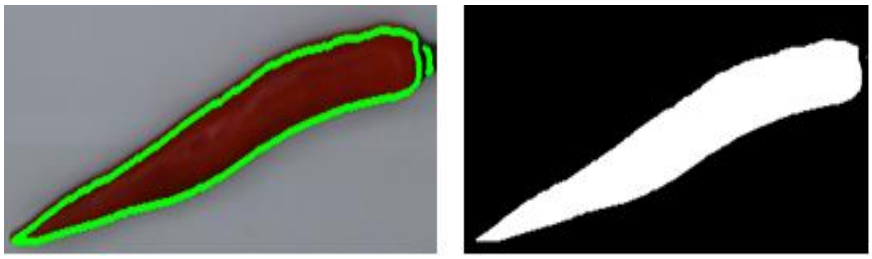
Untuk memperjelas objek pada citra, biasanya operasi *opening* dilakukan terlebih dahulu untuk menghilangkan elemen piksel menonjol atau *noise* dan diikuti dengan operasi *closing* digunakan untuk menghubungkan piksel objek yang terlihat mirip dengan operasi erosi dan dilasi saja, namun kedua operasi tersebut cenderung menghasilkan objek yang lebih jelas dan akurat.

### 2.2.5. Analisis Blob

Analisis blob merupakan teknik yang digunakan untuk menyatakan luas area piksel dari suatu *image* yang menjadi fokus deteksi. Untuk menentukan nilai *blob*, ada beberapa hal yang harus diketahui untuk menghasilkan sebuah *blob* yang optimal (Basri dkk, 2015). Area *foreground* dan *background* dapat diketahui dengan menganalisis posisi objek pada masing-masing *frame* video dalam rentang sampling *frame* tertentu. Apabila terjadi perubahan posisi objek dalam suatu area maka area tersebut dikategorikan sebagai *foreground*. Sedangkan apabila tidak terjadi perubahan posisi objek maka area ini dikategorikan sebagai *background*. Oleh karena itu dapat ditarik kesimpulan bahwa jika objek yang diinginkan melakukan pergerakan dalam video maka objek tersebut akan dapat dikenali.

Proses pemisahan dilakukan dengan memperbaharui nilai parameter *background*. Tahapan awal yaitu mengubah *frame* RGB menjadi biner. Area objek yang terdeteksi sebagai *background* akan ditandai oleh nilai piksel 0 (warna

hitam), sedangkan area objek yang terdeteksi sebagai *foreground* akan ditandai oleh nilai piksel 1 (warna putih). Selanjutnya digunakan fungsi untuk mengenali area piksel yang saling berdekatan/bertetangga, sehingga kumpulan piksel yang termasuk *foreground* akan membentuk satu kesatuan *blob* dari objek yang akan dideteksi (Indrabulan, 2017). Hal yang perlu diperhatikan pada proses ini yaitu tidak semua *blob* yang terbentuk adalah objek yang diinginkan oleh karena itu dibutuhkan filter untuk meminimalisir atau bahkan menghilangkan *blob* ini. Pada Gambar 2.8 menunjukkan proses perubahan citra menjadi blob.



**Gambar 2. 8** Proses perubahan citra menjadi blob

Prosesnya dimulai dari penandaan area *foreground* yang dianggap objek, kemudian pengumpulan data area menjadi blob seperti posisi piksel awal, panjang terhadap sumbu x dan sumbu y dan luas area pada sebuah blob (Basri dkk, 2015).

### 2.2.6. *Region of Interest* (ROI)

*Region of Interest* (ROI) merupakan salah satu proses pengolahan citra dimana pengguna mampu mengolah citra yang mengandung informasi data citra yang dikehendaki. ROI bekerja dalam pengkodean secara berbeda pada area tertentu dari citra digital sehingga kualitas yang lebih baik dari area sekitarnya. Proses ini sangat penting bisa terdapat area tertentu dari citra yang dirasa lebih penting dari bagian lainnya. Area penting tersebut selanjutnya dapat digunakan untuk pengolahan dengan metode tertentu sesuai dengan keperluan penggunaan (Falah dkk., 2016).

Dengan membuat ROI, dapat difokuskan komputasi sesuai dengan posisi yang diinginkan. ROI dibuat beberapa tujuan diantaranya, lokalisasi proses komputasi, meningkatkan akurasi, meminimalisir penggunaan *storage* karena tidak terjadi komputasi pada area yang tidak diinginkan, serta meminimalisir

terjadinya miskomputasi, dan lain sebagainya. Sifat ROI dapat ditemukan pada banyak program khususnya pada proses pengolahan citra (Mufti, 2018).

## 2.3 Data Scaling

Data scaling atau normalisasi data merupakan teknik mengubah nilai numerik dalam dataset ke skala umum, tanpa mendistorsi perbedaan dalam rentang nilai. Normalisasi data akan membantu mempercepat proses pembelajaran pada Machine Learning (Liu, 2011). Pada data hasil ekstraksi fitur dilakukan normalisasi data. Data scaling dibagi menjadi 2 metode yakni:

a. Normalisasi Min-Max

Normalisasi min-max mengubah ukuran data dari rentang asli, sehingga semua nilai berada dalam kisaran 0 dan 1. Persamaan dapat dilihat pada persamaan 2.15 (Liu, 2011).

$$v_{norm} = \left(\frac{v_i - v_{min}}{v_{max} - v_{min}}\right) \tag{2.15}$$

b. Standarisasi (Zero-Mean)

Metode normalisasi Zero-Mean didasarkan pada *mean* dan standar deviasi. Standarisasi suatu dataset melibatkan pengubahan skala distribusi nilai, sehingga nilai rata-rata (*mean*) yang diamati adalah 0 dan standar deviasi adalah 1. Standar deviasi dihitung menggunakan persamaan 2.16 (Liu, 2011).

$$x_{std} = \sqrt{\frac{1}{N-1}\sum_{I=1}^{N}(x_i - x_{mean})^2} \tag{2.16}$$

$x_{mean}$ adalah rata-rata dari data. Standarisasi dapat dihitung dengan persamaan:

$$x_i' = \frac{x_i - x_{mean}}{x_{std}} \tag{2.17}$$

## 2.4 Segmentasi

Segmentasi objek di dalam citra bertujuan memisahkan wilayah (*region*) objek dengan wilayah latar belakang. Selanjutnya, wilayah objek yang telah

tersegmentasi digunakan untuk proses berikutnya (deteksi tepi, pengenalan pola, dan interpretasi objek) (Munir, 2006).

### 2.4.1 Deteksi Tepi dengan Operator Canny

Salah satu algoritma deteksi tepi modern adalah deteksi tepi menggunakan metode Canny. Ada beberapa kriteria pendeteksi tepian paling optimum yang dapat dipenuhi oleh algoritma Canny (Ginting, 2012):

a. Mendeteksi dengan baik (kriteria deteksi)

   Kemampuan untuk meletakkan dan menandai semua tepi yang ada sesuai dengan pemilihan parameter-parameter konvolusi yang dilakukan. Sekaligus juga memberikan fleksibilitas yang sangat tinggi dalam hal menentukan tingkat deteksi ketebalan tepi sesuai yang diinginkan.

b. Melokalisasi dengan baik (kriteria lokalisasi)

   Dengan Canny dimungkinkan dihasilkan jarak yang minimum antara tepi yang dideteksi dengan tepi yang asli.

c. Respon yang jelas (kriteria respon)

   Hanya ada satu respon untuk tiap tepi. Sehingga mudah dideteksi dan tidak menimbulkan kerancuan pada pengolahan citra selanjutnya.

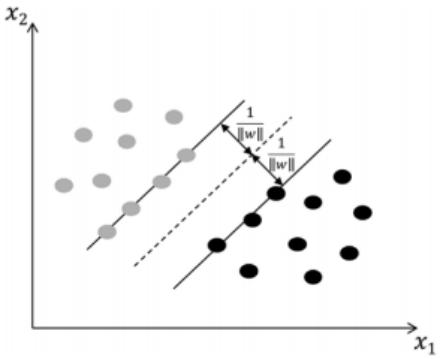
## 2.5 Support Vector Machine (SVM)

SVM merupakan sistem pembelajaran menggunakan ruang berupa fungsi-fungsi linear dalam sebuah ruang fitur yang berdimensi tinggi yang dilatih menggunakan algoritma pembelajaran berdasarkan pada teori optimasi dengan mengimplementasikan *learning bias* (Santosa, 2007)

Pendekatan dengan menggunakan SVM ini memiliki banyak manfaat lain seperti misalnya model yang dibangun memiliki ketergantungan eksplisit pada subset dari data points, serta *support vector* yang membantu dalam interpretasi model. Prinsip utama menggunakan SVM adalah mencari *hyperplane* terbaik yang berfungsi sebagai pemisah dua buah kelas pada ruang *input*. *Hyperplane* tersebut dapat berupa *line* pada *two dimension* dan dapat berupa *flat plane* pada *multiple plane*. SVM merupakan salah satu *machine learning* yang melakukan pelatihan dengan menggunakan *training dataset* dan melakukan generalisasi dan membuat prediksi dari data baru (Kesumawati, 2018).

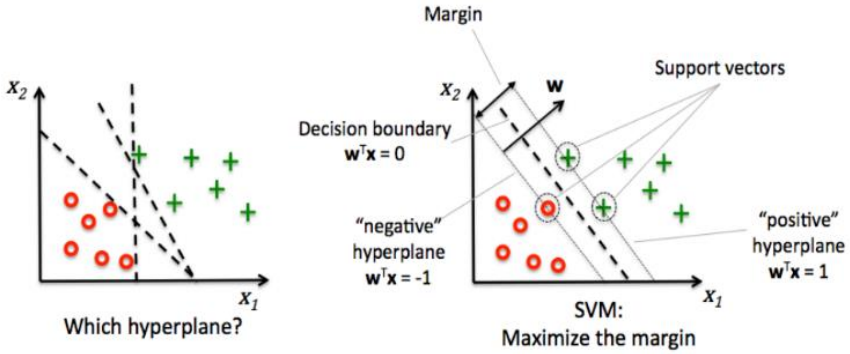
### 2.5.1 *Hard- Margin* SVM / Linear SVM

Teknik SVM merupakan klasifier yang menemukan *hyperplane* dengan kasus data yang digunakan merupakan data dengan dua kelas yang sudah terpisah secara linear seperti pada gambar berikut.




**Gambar 2. 9** *Hard-margin* SVM (Awad & Khanna, 2015)

Berdasarkan pada Gambar 2.9 di atas, terlihat bahwa antara kelas positif dan kelas negatif sudah terpisah secara total terlihat dari lingkaran abu-abu yang berada dekat dengan garis $x_2$ sedangkan untuk lingkaran hitam terletak dekat dengan garis $x_1$ (Awad & Khanna, 2015).




**Gambar 2. 10** *Hyperplane* terbaik yang memisahkan antar dua kelas positif(+1) dan negatif(-1) (Kesumawati, 2018).

Berdasarkan Gambar 2.10 terlihat beberapa pola yang merupakan anggota dari dua belas kelas yaitu positif (+1) dan negatif (-1). *Hyperplane* terbaik dapat ditemukan dengan mengukur *margin hyperplane* dan mencari titik maksimalnya. Margin merupakan jarak antara *hyperplane* dengan data terdekat dari masing-

masing kelas. Subset *training dataset* yang paling dekat dinamakan sebagai *support vector*. Pada Gambar 2.10 sebelah kanan menunjukkan *hyperplane* terbaik, yaitu terletak pada garis putus-putus yang berada tepat ditengah-tengah *hyperplane* positif dan *hyperplane* negatif. Sedangkan tanda "positif" dan "bulat" yang berada dalam lingkaran hitam merupakan *support vector* (Faihah, 2010).

Pencarian lokasi *hyperplane* optimal merupakan inti dari metode SVM. Diasumsikan bahwa terdapat data *learning* dengan *data points* $x_i$ (i = 1, 2,…, m) memiliki dua kelas $y_i = \pm 1$ yaitu kelas positif (+1) dan kelas negatif (-1) sehingga akan diperoleh *decision function* berikut.

$$f(x) = sign\,(w.x + b) \tag{2.18}$$

Dimana (.) merupakan skalar sehingga $w.x \equiv x^T x$

Berdasarkan pada *decision function* di atas, dapat terlihat bahwa data akan terklasifikasi secara tepat jika $y_i(w.x_i + b) > 0\ \forall_i$ karena ketika $(w.x_i + b)$ harus bernilai positif saat $y_i$ = +1, dan bernilai negatif ketika $y_i$ = -1. *Decision function* menjadi invarian ketika akan dilakukan pembuatan skala positif baru dari argumen persamaan fungsi sehingga akan mengakibatkan ambiguitas dalam mendefinisikan konsep jarak atau margin. Maka dari itu didefinisikan skala untuk (*w*,*b*) dengan menetapkan *w.x* + *b* = 1 untuk titik terdekat pada satu sisi dan *w.x*+*b* = 1 dan *w.x*+*b* = -1 disebut sebagai *hyperplane* kanonik dan wilayah antar *hyperplane* disebut sebagai *margin band* (Campbell & Ying, 2011).

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.

$$\tau\,(w) = \frac{1}{2}\|w\|^2\ (3.2) \tag{2.19}$$

Sedangkan subjek *constraint* atau kendala persamaannya adalah sebagai berikut.

$$y_i(w.x_i + b) > 0\ \forall_i \tag{2.20}$$

Persamaan di atas merupakan permasalahan optimasi kendala dimana meminimalkan fungsi objek pada persamaan (2.19) dengan kendala pada persamaan (2.20). Permasalahan di atas dapat direduksi dengan menggunakan *Lagrange* yang terdiri dari jumlahan fungsi objektif dan m kendala dikalikan dengan pengganda *Lagrange* seperti berikut (Campbell & Ying, 2011).

$$L(w,b) = \frac{1}{2}(w.w) - \sum_{i=1}^{m} \alpha_i(y_i(w.x_i + b) - 1) \quad (2.21)$$

Dimana $\alpha_i$ 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.

$$\frac{\partial L}{\partial b} = -\sum_{i=1}^{m} \alpha_i y_i = 0 \quad (2.22)$$

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

Substitusi nilai *w* dari persamaan (2.23) kedalam bentuk L(w, b) sehingga akan diperoleh rumus ganda atau biasa disebut sebagai *wolfe dual*.

$$W(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j (x_i x_j) \quad (2.24)$$

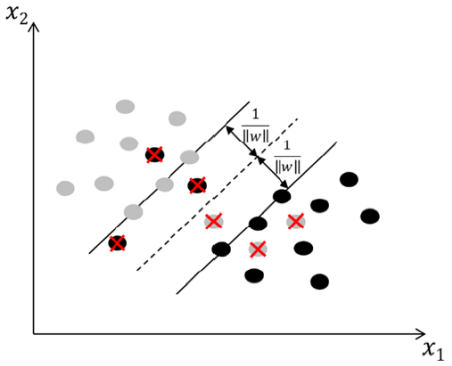
Dimana nilai $\alpha_i$ terhadap kendala adalah sebagai berikut.

$$\alpha_i \geq 0 \qquad \sum_{i=1}^{m} \alpha_i y_i \quad (2.25)$$

### 2.5.2 *Soft –Margin* SVM

Ketika data yang digunakan tidak sepenuhnya dapat dipisahkan, *slack variables* $x_i$ diperkenalkan ke dalam fungsi objektif 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 (Awad & Khanna, 2015).




**Gambar 2. 11** Beberapa misklasifikasi pada *Soft margin* SVM (Awad & Khanna, 2015)

Berdasarkan pada Gambar 2.11, 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 variabel* pada persamaan (2.20) sebelumnya seperti berikut.

$$y_i(w.x_i + b) > 1 - \varepsilon_i \tag{2.26}$$

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

$$\left[\frac{1}{2} w.w + C \sum_{i=1}^{m} \varepsilon_i\right] - \sum_{i=1}^{m} \alpha_i [y_i(w.x_i + b) - 1 + \varepsilon_i] - \sum_{i=1}^{m} r_i \varepsilon_i \tag{2.27}$$

Dengan demikian, persamaan (2.19) akan diubah kedalam persamaan berikut.

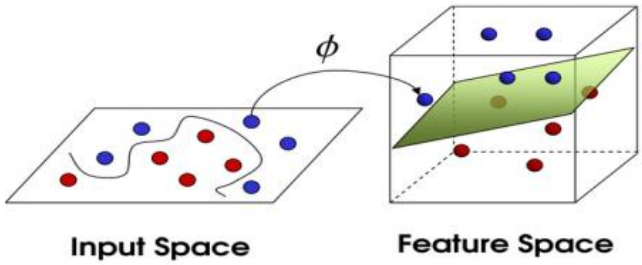
$$\tau(w) = \frac{1}{2} \|w\|^2 + c \sum_{i=1}^{m} \varepsilon_i \tag{2.28}$$

Parameter c digunakan untuk mengontrol *trade off* antara margin dan kesalahan klasifikasi $\sum$ (Abtohi, 2017).

### 2.5.3 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 (Awad & Khanna, 2015). Berikut kernel SVM untuk memisahkan data secara *linear* ditunjukkan pada Gambar 2.12.




**Gambar 2.12** Kernel SVM untuk memisahkan data secara *linear* (Kesumawati, 2018)

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 (Campbell and Ying, 2011). 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 (Rai, 2011).

$$\boldsymbol{K}(\boldsymbol{x},\boldsymbol{z}) = (\boldsymbol{x}^{\boldsymbol{T}}\boldsymbol{z})^2$$
$$\boldsymbol{K}(\boldsymbol{x},\boldsymbol{z}) = (x_1 z_1 + x_2 z_2)^2$$
$$\boldsymbol{K}(\boldsymbol{x},\boldsymbol{z}) = ({x_1}^2 {z_1}^2 + {x_2}^2 {z_2}^2 + 2x_1 x_2 z_1 z_2)^2$$
$$\boldsymbol{K}(\boldsymbol{x},\boldsymbol{z}) = \left({x_1}^2, \sqrt{2}x_1 x_2, {x_2}^2\right)^T \left({z_1}^2, \sqrt{2}z_1 z, {z_2}^2\right)$$
$$\boldsymbol{K}(\boldsymbol{x},\boldsymbol{z}) = \Phi(x)^T \Phi(z) \qquad (2.29)$$

Nilai **K** di atas secara implisit mendefinisikan pemetaan ke ruang dimensi yang lebih tinggi seperti berikut.

$$\Phi(x) = \{{x_1}^2, \sqrt{2}x_1x_2, {x_2}^2\} \tag{2.30}$$

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

$$\Phi : X \rightarrow F$$
$$\boldsymbol{K} : X\ x\ X \rightarrow R, \boldsymbol{K}(\boldsymbol{x}, \boldsymbol{z}) = \ \Phi(x).\Phi(z)$$

Berdasarkan pada fungsi kernel di atas, dapat dilakukan perhitungan untuk melakukan prediksi dari beberapa data dalam *feature space* seperti pada persamaan berikut (Cambell & Ying, 2011).

$$f\big(\Phi(x)\big) = sign(w.\Phi(z) + b)$$
$$f\big(\Phi(x)\big) = sign(\textstyle\sum_{i=1}^{m} \alpha_i y_i \boldsymbol{K}(\boldsymbol{x}, \boldsymbol{z}) + b) \tag{2.31}$$

Keterangan:

b : nilai bias

m : jumlah *support vector*

**K(x,z)** : fungsi kernel

Nilai k yang bisa digunakan sebagai fungsi kernel harus memenuhi kondisi Mercer antara lain (Rai, 2011):

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

$$\int dx \int dz f(x)\mathbf{K}(\mathbf{x}, \mathbf{z})f(z) > 0 \qquad (\forall f \in L_2) \tag{2.32}$$

- Ketika $k_1$ dan $k_2$ merupakan fungsi kernel, maka:

$$\mathbf{K}(\mathbf{x}, \mathbf{z}) = \mathbf{K_1}(\mathbf{x}, \mathbf{z}) + \mathbf{K_2}(\mathbf{x}, \mathbf{z}) : Direct\ sum \tag{2.33}$$
$$\mathbf{K}(\mathbf{x}, \mathbf{z}) = \alpha\mathbf{K_1}(\mathbf{x}, \mathbf{z}) : Skalar\ Product \tag{2.34}$$
$$\mathbf{K}(\mathbf{x}, \mathbf{z}) = \mathbf{K_1}(\mathbf{x}, \mathbf{z})\mathbf{K_2}(\mathbf{x}, \mathbf{z}) : Direct\ Product \tag{2.35}$$

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 (Kowalczyk, 2014). Berikut merupakan persamaan dari linear kernel SVM.

$$\boldsymbol{K}(\boldsymbol{x},\boldsymbol{z}) = \boldsymbol{x}^{\boldsymbol{T}}\boldsymbol{z} \tag{2.36}$$

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.

$$\boldsymbol{K}(\boldsymbol{x},\boldsymbol{z}) = (\boldsymbol{x}^{\boldsymbol{T}}\boldsymbol{z})^{d}\ atau\ (1+\boldsymbol{x}^{\boldsymbol{T}}\boldsymbol{z})^{d} \tag{2.37}$$

**3.** *Radian 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.* Semakin tinggi nilai C, maka kemungkinan terjadinya kesalahan dalam penentuan solusi akan semakin kecil. Sebaliknya, jika nilai C semakin rendah maka semakin tinggi proporsi kesalahan yang terjadi pada penentuan solusi (Sari dkk, 2017). 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 disekitar garis yang masuk akal akan dipertimbangkan dalam perhitungan (Patel, 2017). Berikut persamaan dari RBF kernel.

$$\boldsymbol{K}(\boldsymbol{x}, \boldsymbol{z}) = \exp[-\gamma \|\boldsymbol{x} - \boldsymbol{z}\|^2 \quad (2.38)$$

Pemilihan fungsi kernel yang tepat merupakan hal yang sangat penting karena akan menentukan *feature space* dimana fungsi *classifier* akan dicari. Sepanjang fungsi kernelnya sesuai (cocok), SVM akan beroperasi secara benar meskipun tidak tahu pemetaan yang digunakan (Santosa, 2007). Menurut (Scholkopf & Smola, 2002), fungsi kernel Gaussian RBF memiliki kelebihan yaitu secara otomatis menentukan nilai, lokasi dari *center* dan nilai pembobot dan bisa mencakup nilai rentang tak terhingga. Gaussian RBF juga efektif menghindari *overfitting* dengan memilih nilai yang tepat untuk parameter C dan γ dan RBF baik digunakan ketika tidak ada pengetahuan terdahulu. Menurut (Hsu dkk, 2002), fungsi kernel yang direkomendasikan untuk diuji pertama kali adalah fungsi kernel RBF karena dapat memetakan hubungan tidak linear RBF lebih *robust* terhadap *outlier* karena fungsi kernel RBF berada antara selang (-∞, ∞) sedangkan fungsi kernel yang lain memiliki rentang antara (-1 sampai dengan 1).

Beberapa keunggulan Support Vector Machine antara lain:

a. SVM efektif pada data berdimensi tinggi (data dengan jumlah fitur atau atribut yang sangat banyak) dan efektif pada kasus dimana jumlah fitur pada data lebih besar dari jumlah sampel.
b. SVM menggunakan subset poin pelatihan dalam fungsi keputusan (disebut *support vector*) sehingga membuat penggunaan memori menjadi lebih efisien.

### 2.5.4 *Multi Class* 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 2.23 (Pricila, 2016).

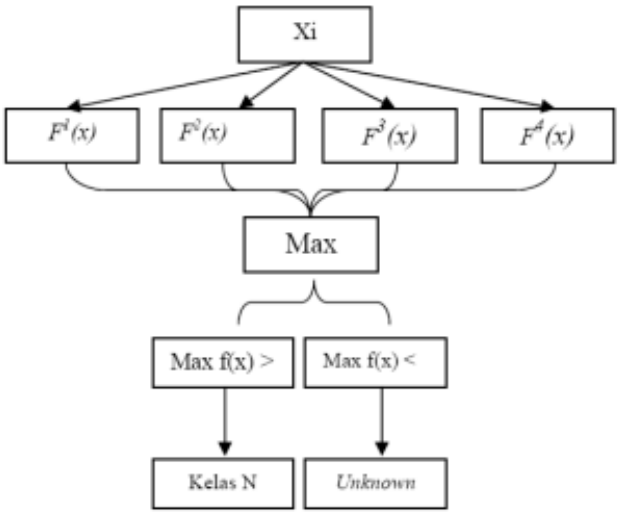
$$\begin{aligned} &\min_{w^i, b^i, \xi^i} \frac{1}{2}(w^i)^T w^i + C\sum_t \xi_t^i \\ &s.t\ \left(w^i\right)^T \phi(x_t) + b^i \geq 1 - \xi_t^i \rightarrow y_t = i, \\ &\left(w^i\right)^T \phi(x_t) + b^i \geq -1 + \xi_t^i \rightarrow y_t \neq i, \\ &\xi_t^i \geq 0 \end{aligned} \tag{2.39}$$

Contohnya, terdapat permasalahan klasifikasi dengan 4 buah kelas. Untuk pelatihan digunakan 4 buah SVM biner seperti pada Tabel 2.2 (Pricila, 2016).

**Tabel 2. 2** 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 gambar 2.13.




**Gambar 2. 13** 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:

$$Kelas\ g = \arg\max_{r=1,..,z}\left(\left(w^{(r)}\right)^{T}.\varphi\begin{bmatrix}p_i\\q_i\end{bmatrix} + b^{(r)}\right) \qquad (2.40)$$

Dimana:

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

$\varphi\begin{bmatrix}p_i\\q_i\end{bmatrix}$ = *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 2.25 (Pricila, 2016).

$$\min_{w^{ij}, b^{ij}, \xi^{ij}} \frac{1}{2}(w^{ij})^T w^{ij} + C\sum_t \xi_t^{ij}$$

$$s.t\left(w^{ij}\right)^T \phi(x_t) + b^{ij} \geq 1 - \xi_t^{ij} \rightarrow y_t = i,$$

$$\left(w^{ij}\right)^T \phi(x_t) + b^{ij} \geq -1 + \xi_t^{ij} \rightarrow y_t = j,$$

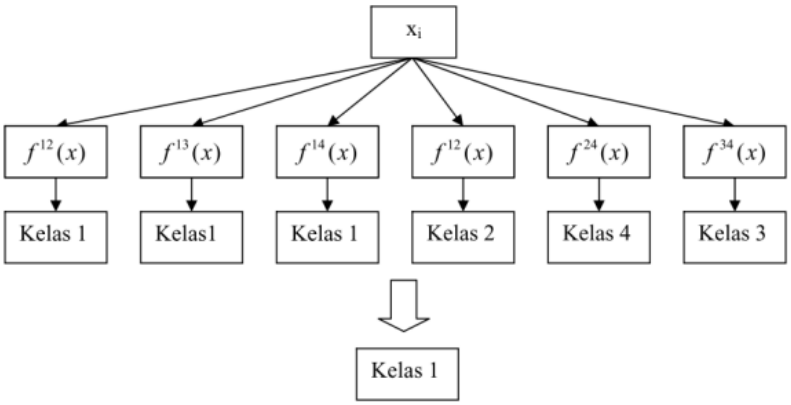
$$\xi_t^{ij} \geq 0 \qquad (2.41)$$

Jika data x dimasukkan ke dalam fungsi hasil pelatihan $f(x) = \left(w^{ij}\right)^T \phi(x) + b$ 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 indeksnya 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 2.3.

**Tabel 2. 3** 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 Gambar 2.14.




**Gambar 2. 14** Contoh klasifikasi dengan metode One-Against-One (Pricila, 2016)

## 2.6 Penelitian Terkait

### 2.6.1 Klasifikasi Kualitas Tomat Buah Menggunakan *Video Processing* (Riny Yustica Dewi, 2021)

Penelitian ini dilakukan dengan menggunakan data video untuk masing-masing kualitas tomat buah. Peneliti membagi kualitas tomat buah menjadi 2 kelas, yakni kelas tomat buah layak dan tidak layak. Dataset yang digunakan terdiri dari 30 tomat buah untuk *training* dan 15 tomat buah untuk *testing* pada setiap kategorinya. Penentuan kualitas tomat buah diidentifikasi berdasarkan warna dengan memanfaatkan fitur *color* moment seperti mean dan *standard* deviation pada *channel* RGB, sistemnya juga menggunakan *Hue, Saturation,* Value (HSV) untuk mengelompokkan warna. Metode yang digunakan untuk klasifikasi adalah *Grid Search* SVM dengan parameter C = 0.1 dan kernel *linear*. Hasil akurasi yang didapatkan yaitu sebesar 95.55%.

### 2.6.2 Prototype Sistem Klasifikasi Kematangan Stroberi Menggunakan Algoritma SVM (Nurhikmah Arifin, 2019)

Penelitian ini dilakukan menggunakan data berupa video untuk masing-masing kelas kematangan stroberi. Dimana peneliti membagi

stroberi menjadi 4 kelas, yaitu kelas stroberi belum matang, setengah matang, matang, dan busuk. Data yang digunakan terdiri dari 70 buah stroberi untuk data latih dan 25 buah stroberi data uji untuk masing-masing kategori. Hasil penelitian ini menunjukkan bahwa klasifikasi kematangan stroberi menggunakan algoritma *multi class* SVM dengan parameter kernel RBF, *Cost* (C) = 10 dan gamma ($\gamma$) = $10^{-3}$ menghasilkan akurasi tinggi yaitu 90,31%.

### 2.6.3 *The Development of Machine Vision System for Sorting Passion Fruit using Multi Class Support Vector Machine* (Sidehabi, Sitti Wetenriajeng, et al, 2018)

Penelitian ini mengidentifikasi tingkat kematangan buah markisa. Markisa memiliki tingkat kematangan yang bervariasi maka dengan memanfaatkan teknologi kecerdasan buatan, sistem ini mengklasifikasikan buah markisa menjadi 3 bagian yaitu matang, hampir matang, dan belum matang. Data yang diolah berupa video dengan mengambil 6 sisi buah markisa sehingga data yang diproses untuk satu markisa adalah 6 *frame* per video. Metode yang digunakan untuk segmentasi adalah *K-Means Cluster*, sedangkan ekstraksi fitur menggunakan nilai RGB*a dengan total fitur yang digunakan adalah 24 fitur yaitu 4 fitur RGB*a x 6 sisi buah markisa dan *Multi-class Support Vector Machine* (SVM). Hasil uji coba menunjukkan bahwa klasifikasi terbaik menggunakan kernel RBF dengan akurasi sebesar 93,9%. Kesalahan klasifikasi terjadi pada buah yang hampir matang dan belum matang karena kedekatan warna. Kernel yang digunakan adalah kernel RBF dengan parameter C = 25 dan $\gamma = 10^{-5}$ dengan rata-rata deteksi untuk tiap buahnya adalah 0.94128/detik.

### 2.6.4 Deteksi Kematangan Buah Pisang Berdasarkan Fitur Warna Citra Kulit Pisang Menggunakan Metode Transformasi Ruang Warna HIS (Indarto, Murinto, 2017)

Penelitian ini mendeteksi kematangan buah pisang berdasarkan fitur warna citra kulit pisang. Data yang diolah merupakan citra pisang ambon yang diambil dengan kamera dan dilakukan *cropping* pada gambar. Hasil

*cropping* kemudian diekstrak ciri warnanya, dan dihitung tingkat kadar warna dari RGB dan diubah ke HSI. Berdasarkan *input* pelatihan deteksi jenis warna kulit pisang ambon diperoleh dari pengolahan citra dengan metode transformasi warna HSI. Dari hasil penelitian 20 sampel buah terdiri dari 10 buah pisang ambon mentah dan 10 buah pisang ambon matang dengan nilai rata-rata maksimal dan minimal H, S, I diperoleh akurasi kesesuaian sebesar 85%.