

**SKRIPSI**

**KLASIFIKASI KUALITAS TOMAT BUAH MENGGUNAKAN *VIDEO*  
*PROCESSING***

**Disusun dan diajukan oleh**

**RINY YUSTICA DEWI**

**D421 16 010**



**DEPARTEMEN TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERISTAS HASANUDDIN**

**MAKASSAR**

**2021**

**LEMBAR PENGESAHAN SKRIPSI**

**KLASIFIKASI KUALITAS TOMAT BUAH MENGGUNAKAN *VIDEO*  
*PROCESSING***

Disusun dan diajukan oleh

**RINY YUSTICA DEWI**

**D421 16 010**

Telah dipertahankan di hadapan Panitia Ujian yang dibentuk dalam rangka penyelesaian studi Program Sarjana Program Studi Teknik Informatika Fakultas Teknik Universitas

Hasanuddin

Pada tanggal 5 Februari 2021  
dan dinyatakan telah memenuhi syarat kelulusan

Menyetujui

Pembimbing Utama,

Pembimbing Pendamping,

**Dr. Indrabayu, S.T., M.T., M.Bus.Sys.**  
NIP. 19750716 200212 1 004

**Dr. Eng. Intan Sari Areni, S.T., M.T.**  
NIP. 19750203 200012 2 002

Ketua Program Studi,



**Dr. Amil Ahmad Ilham, S.T., M.IT.**  
NIP. 19711010 199802 1 001

## PERNYATAAN KEASLIAN

Yang bertanda tangan di bawah ini:

Nama : RINY YUSTICA DEWI

NIM : D421 16 010

Program Studi : Teknik Informatika

Jenjang : S1

Menyatakan dengan ini bahwa karya tulisan saya berjudul:

### **KLASIFIKASI KUALITAS TOMAT BUAH MENGGUNAKAN *VIDEO* *PROCESSING***

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

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.

Makassar, 5 Februari 2021

Yang Menyatakan,



**RINY YUSTICA DEWI**

## KATA PENGANTAR

Assalamu'alaikum Warahmatullaahi Wabarakatuh.

Alhamdulillahilahirabbil'aalamiin, puji syukur senantiasa penulis panjatkan atas kehadiran Allah SWT yang telah melimpahkan rahmat dan hidayah-Nya yang tak terhingga, sehingga penulis dapat menyelesaikan tugas akhir ini yang berjudul **“Klasifikasi Kualitas Tomat Buah Menggunakan *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 InsyaAllah selalu tercurahkan kepada junjungan kita Nabi Muhammad SAW yang telah menuntun kita dari jalan yang gelap menuju yang terang yakni Dienul Islam.

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. Oleh karena itu, penulis dengan senang hati menyampaikan terima kasih kepada:

1. Kedua Orang Tua penulis, Bapak Muhammad Sabrang, S.E., M.M. dan Ibu Ariyanty, S.Pd. yang selalu memberikan dukungan, doa, dan semangat serta selalu sabar dalam mendidik penulis sejak kecil.
2. Bapak Dr. Indrabayu, S.T., M.T., M.Bus.Sys., selaku pembimbing I dan Ibu Dr. Eng. Intan Sari Areni, S.T. M.T., selaku pembimbing II yang selalu menyediakan waktu, tenaga, pikiran, dan perhatian yang luar biasa untuk mengarahkan penulis dalam penyusunan tugas akhir ini.

3. Bapak Dr. Amil Ahmad Ilham, S.T., M.I.T., selaku Ketua Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin atas bimbingannya selama masa perkuliahan penulis.
4. Teman-teman dan kakak-kakak *AIMP Research Group* FT-UH yang telah memberikan begitu banyak bantuan selama penelitian, pengambilan data, dan diskusi terkait progres penyusunan Tugas Akhir.
5. Kak Nur Hikmah yang telah meminjamkan *belt conveyor* yang digunakan dalam pengambilan data penelitian.
6. Keluarga besar dari aLias yang senantiasa memberikan semangat dan juga motivasi selama pengerjaan Tugas Akhir.
7. Andi Amelia Ramadanti dan Andi Zavira Ardhia, A. N yang memberi dukungan dalam penyelesaian Tugas Akhir.
8. Saudara-saudara Igniter16 FT-UH atas dukungan dan semangat yang diberikan selama ini.
9. Segenap staf dan Dosen Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin yang telah membantu penulis.
10. Serta seluruh pihak yang tak sempat penulis sebutkan satu persatu yang telah banyak meluangkan tenaga, waktu, dan pikiran 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, Desember 2020

Penulis

## ABSTRAK

Peningkatan permintaan dan produksi tomat oleh masyarakat nasional maupun internasional yang semakin meningkat menjadikan tomat memiliki nilai ekonomi yang tinggi dan masih memerlukan penanganan serius, terutama dalam hal peningkatan kualitas buahnya. Proses penentuan kualitas tomat buah umumnya dilakukan secara manual bergantung pada persepsi manusia sehingga memiliki kelemahan dan keterbatasan. Menurut Standar Nasional Indonesia, syarat penentuan mutu dalam pengolahan tomat buah dari segi warna yaitu “normal”. Namun kata “normal” memiliki arti yang subjektif sehingga setiap orang memiliki persepsi yang berbeda-beda terhadap kata tersebut. Oleh karena itu, penelitian ini bertujuan untuk membantu dalam mengklasifikasikan kualitas tomat buah menggunakan metode *Grid Search SVM* berbasis *video processing* dengan kernel *linear* dan membagi tomat buah menjadi dua kategori kelas, yaitu kelas 1 untuk tomat buah layak dan kelas 2 untuk tomat buah tidak layak. Data diambil menggunakan *Logitech Webcam C922 Pro*, dataset tersebut terdiri dari 30 tomat buah untuk *training* dan 15 tomat buah untuk *testing* pada setiap kategorinya dengan ukuran *frame* 960 x 540 *pixel* dan dilakukan proses *Gaussian Filter* pada tahap *preprocessing*. Penentuan kualitas tomat buah diidentifikasi berdasarkan warna dengan memanfaatkan fitur *color moment* seperti *mean* dan *standard deviation* pada *channel* RGB, sistem ini juga menggunakan *Hue*, *Saturation*, *Value* (HSV) untuk mengelompokkan warna. Pada penelitian ini diperoleh akurasi sistem sebesar 95.55% dengan parameter *C* optimal yang didapatkan menggunakan *Grid Search SVM* yaitu 0.1.

**Kata kunci:** Tomat buah, HSV, *linear*, *Grid Search SVM*

## DAFTAR ISI

<b>HALAMAN JUDUL</b> .....	i
<b>LEMBAR PENGESAHAN SKRIPSI</b> .....	ii
<b>PERNYATAAN KEASLIAN</b> .....	iii
<b>KATA PENGANTAR</b> .....	iv
<b>ABSTRAK</b> .....	vii
<b>DAFTAR ISI</b> .....	viii
<b>DAFTAR GAMBAR</b> .....	x
<b>DAFTAR TABEL</b> .....	xi
<b>BAB I PENDAHULUAN</b> .....	1
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	5
1.3 Tujuan Penelitian.....	5
1.4 Manfaat Penelitian.....	5
1.5 Batasan Masalah Penelitian.....	6
1.6 Sistematika Penulisan.....	6
<b>BAB II TUJUAN PUSTAKA</b> .....	8
2.1 Tomat .....	8
2.2 Pengolahan Citra .....	9
2.3 Visi Komputer .....	14
2.4 <i>Gaussian Filter</i> .....	17



2.5	<i>RGB Color Channel</i> .....	18
2.6	Segmentasi Warna Menggunakan <i>Hue, Saturation, Value</i> (HSV) .....	19
2.7	<i>Thresholding</i> .....	21
2.8	<i>Blob Detection</i> .....	22
2.9	<i>Support Vector Machine</i> (SVM) .....	23
2.10	<i>Cross Validation and Grid Search SVM</i> .....	34
2.11	Penelitian Terkait .....	35
<b>BAB III METODOLOGI PENELITIAN</b> .....		39
3.1	Tahap Penelitian .....	39
3.2	Waktu dan Lokasi Penelitian.....	40
3.3	Instrumen Penelitian.....	40
3.4	Teknik Pengambilan Data .....	41
3.5	Rancangan Sistem .....	42
3.6	Analisis Kerja Sistem .....	58
<b>BAB IV HASIL DAN PEMBAHASAN</b> .....		59
4.1	Hasil Penelitian .....	59
4.2	Pembahasan .....	61
<b>BAB V PENUTUP</b> .....		65
5.1	Kesimpulan.....	65
5.2	Saran.....	66
<b>DAFTAR PUSTAKA</b> .....		67
<b>LAMPIRAN</b> .....		71

## DAFTAR GAMBAR

Gambar 2.1 Ruang Warna HSV.....	21
Gambar 2.2 Area <i>Blob</i> .....	23
Gambar 2.3 Ilustrasi Pencarian <i>Hyperplane</i> Terbaik Menggunakan SVM.....	24
Gambar 2.4 Beberapa Misklasifikasi Pada <i>Soft Margin</i> SVM .....	27
Gambar 2.5 Kernel SVM Untuk Memisahkan Data Secara <i>Linear</i> .....	29
Gambar 3.1 Tahap Penelitian.....	39
Gambar 3.2 Skenario Pengambilan Data .....	42
Gambar 3.3 Alur Perancangan Sistem Klasifikasi Kualitas Tomat Buah.....	43
Gambar 3.4 <i>Frame</i> Tomat Buah Pada Proses <i>Training</i> .....	45
Gambar 3.5 <i>Frame</i> Tomat Buah Hasil <i>Resize</i> .....	45
Gambar 3.6 <i>Frame</i> Tomat Buah Hasil Perbaikan Kualitas Citra .....	46
Gambar 3.7 Nilai Matriks Hasil Konversi RGB ke HSV .....	48
Gambar 3.8 <i>Frame</i> Tomat Buah Hasil Konversi RGB ke HSV .....	48
Gambar 3.9 <i>Frame</i> Tomat Buah Pada Proses <i>Masking</i> .....	49
Gambar 3.10 <i>Frame</i> Tomat Buah Hasil <i>Bounding Box</i> dan Hasil <i>Crop</i> .....	49
Gambar 3.11 Alur Sistem Proses Pelatihan <i>Grid Search</i> SVM.....	52
Gambar 3.12 Model <i>Training Grid Search</i> SVM dengan Kernel <i>Linear</i> .....	55
Gambar 3.13 <i>Frame</i> Tomat Buah Hasil Klasifikasi Tomat Buah .....	58
Gambar 4.1 <i>Confusion Matrix</i> .....	60

## DAFTAR TABEL

Tabel 3.1 Kriteria Tomat Buah .....	44
Tabel 3.2 <i>Range</i> Nilai HSV .....	49
Tabel 3.3 <i>Range</i> Nilai RGB .....	50
Tabel 3.4 <i>Support Vector</i> Masing-Masing Kelas.....	56
Tabel 4.1 Hasil Akurasi Sistem dengan Parameter $C = 0.1$ dan Kernel <i>Linear</i> ....	60
Tabel 4.2 Tabel Hasil Kesalahan Klasifikasi Karena Fitur Warna RGB.....	62
Tabel 4.3 Tabel <i>Frame</i> Hasil Kesalahan Klasifikasi .....	63

# **BAB I**

## **PENDAHULUAN**

### **I.1 Latar Belakang**

Industri pertanian saat ini berkembang dengan pesat. Indonesia merupakan salah satu negara yang memiliki potensi besar dalam bidang pertanian. Iklimnya yang tropis dengan curah hujan yang tinggi sepanjang tahun serta tanah yang subur, memungkinkan tumbuhnya berbagai jenis tanaman (Shafira, 2018). Salah satu dari sekian banyak jenis tanaman potensial yang tumbuh di Indonesia ialah tanaman hortikultura. Kebutuhan masyarakat Indonesia akan tanaman hortikultura sangat tinggi. Salah satu jenis hasil hortikultura yang disukai oleh masyarakat adalah tomat. Tomat (*Solanum lycopersicum L.*) merupakan komoditas hortikultura yang penting dalam memenuhi kebutuhan pasar dan masyarakat. Pemanfaatan semakin banyak, selain dikonsumsi sebagai tomat segar dan bumbu masakan, buah tomat dapat diolah menjadi bahan dasar industri makanan seperti sari buah dan saus tomat (Choulillah F., 2016).

Peningkatan permintaan dan produksi tomat oleh masyarakat nasional maupun internasional yang semakin meningkat (Shafira, 2018), menjadikan tomat memiliki nilai ekonomi yang tinggi dan masih memerlukan penanganan serius, terutama dalam hal peningkatan kualitas buahnya (Firmansyah, 2016). Berdasarkan data dari Badan Pusat Statistik dan Direktorat Jenderal Hortikultur menyebutkan bahwa tomat menempati urutan kelima yang memiliki tingkat produktivitas yang tinggi di Sulawesi Selatan pada tahun 2018 sebesar 67.374 (Kementerian Pertanian Republik Indonesia, 2018). Sehingga pengembangan dan penerapan teknologi

produksi olahan tomat perlu terus dipacu dalam upaya meningkatkan kualitas produk sehingga mampu meningkatkan pendapatan industri kecil dan menengah (Abbas & Suhaeti, 2016). Penanganan pemilihan untuk produksi olahan memegang peranan penting dalam penentuan kualitas tomat. Selama ini kegiatan penentuan kualitas tomat buah dilakukan secara manual, sehingga menghasilkan produk yang kurang seragam (Zaenury Ichsan, Andrizal, & Yendri, 2014). Dalam industri kecil dan menengah proses pemilihan atau penyortiran umumnya sangat bergantung pada persepsi manusia terhadap komposisi warna, ukuran, bentuk, serta terdapatnya cacat atau kebusukan dan rasa produk pertanian berdasarkan kekerasan (tekstur) dan manis. Warna buah dan sayur merupakan parameter utama dari konsumen untuk menentukan kualitas. Warna faktor utama menentukan kualitas dan menjadi sebagai atribut sensori yang dapat diamati langsung sebagai indikator kesegaran dan kematangan, sebagai contoh tomat akan terjadi perubahan warna dari hijau ke merah. Karakteristik warna tomat adalah yang paling penting untuk menilai kematangan dan pengolahan pascapanen, konsumen secara visual sering menilai kematangan tomat berdasarkan warna (Abdillah, 2019). Cara manual dilakukan berdasarkan pengamatan visual secara langsung pada buah yang akan diklasifikasi. Kelemahan klasifikasi buah secara manual sangat dipengaruhi oleh subjektivitas petugas penyortiran sehingga dalam kondisi tertentu tidak spesifik proses pengklasifikasikannya dan juga dapat mengakibatkan tidak konsistensi dalam proses pemilihannya. Identifikasi dengan cara ini memiliki beberapa kelemahan diantaranya adalah waktu yang dibutuhkan relatif lama serta menghasilkan produk

yang beragam karena adanya keterbatasan visual manusia, tingkat kelelahan dan perbedaan persepsi tentang kualitas buah (Kusumaningtyas & Asmara, 2016).

Karena hasil seleksi manual memiliki beberapa kelemahan, maka diperlukan sebuah sistem yang dapat membantu dalam menyeleksi dan mengelompokkan kualitas tomat buah dengan baik berbasis *video processing*. Adanya perkembangan sistem pengolahan citra ini, memungkinkan dilakukannya penentuan tingkat kualitas tomat buah tanpa merusaknya. Menurut Standar Nasional Indonesia, syarat penentuan mutu dalam pengolahan tomat buah dari segi warna yaitu “normal” (Indonesia & Nasional, 1998). Namun kata “normal” memiliki arti yang subjektif sehingga setiap orang memiliki persepsi yang berbeda-beda terhadap kata tersebut.

Oleh karena itu, pada penelitian ini penulis akan membagi tomat buah menjadi dua kelas yaitu tomat buah layak dan tomat buah tidak layak dengan beberapa kriteria pada setiap kelasnya yang dapat memudahkan dalam penentuan kualitas tomat buah. Dalam penelitian ini algoritma yang digunakan adalah algoritma *Support Vector Machine* (SVM), karena algoritma *Support Vector Machine* (SVM), merupakan algoritma yang sangat populer digunakan untuk permasalahan klasifikasi, melakukan beberapa transformasi data yang sangat kompleks, lalu mencari cara untuk memisahkan data berdasarkan label atau keluaran yang telah ditentukan (Ritonga & Purwaningsih, 2018). Dalam pemodelan klasifikasi, *Support Vector Machine* (SVM), mempunyai konsep yang lebih matang serta lebih jelas secara matematis dibandingkan dengan teknik-teknik klasifikasi yang lain. Namun algoritma ini memiliki kelemahan yaitu mengalami kesulitan

dalam menentukan nilai parameter yang optimal. Yenaeng, Saelee dan Samai (2014) menyatakan bahwa permasalahan terbesar dalam mengatur model SVM adalah menentukan nilai hyperparameter dari SVM. Padahal, pengaturan nilai parameter secara tepat akan meningkatkan akurasi klasifikasi dari model SVM (Huang & Wang, 2006). Untuk mendapatkan parameter yang akan menghasilkan model SVM yang paling baik, maka dilakukan optimasi parameter pada model SVM (Kusumaningrum, 2017). Optimasi parameter tersebut berarti menentukan hyperparameter model SVM yang paling optimal dan menghasilkan model SVM dengan hasil klasifikasi yang paling baik. Metode *Grid Search SVM* merupakan metode yang paling banyak digunakan untuk optimasi parameter (Chen, Ling, Tang & Xia, 2016). Oleh karena itu, pada penelitian ini metode *Grid Search SVM* akan digunakan untuk mengoptimalkan nilai parameter pada model SVM, sehingga dengan parameter yang optimal tersebut diharapkan dapat meningkatkan akurasi hasil klasifikasi yaitu dengan menggunakan kernel *linear*. Dimana pada umumnya kernel linear digunakan untuk mengklasifikasikan data menjadi dua kelas. Melatih SVM dengan kernel *linear* lebih cepat dibandingkan dengan kernel lainnya dan saat melatih SVM dengan kernel linear, hanya pengoptimalan parameter  $C$  regulasi yang diperlukan (Adhani, Buono, & Faqih, 2013). Parameter  $C$  adalah parameter yang menentukan besar kesalahan dalam klasifikasi data dan nilainya ditentukan oleh pengguna. Parameter  $C$  dipilih untuk mengontrol *trade off* antara *margin* dengan kesalahan klasifikasi  $\epsilon$ . Semakin besar nilai  $C$  maka *margin* dalam model SVM akan semakin kecil Nugroho dkk., 2013).

## **I.2 Rumusan Masalah**

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

1. Bagaimana pengaruh parameter  $C$  pada metode *Grid Search SVM* terhadap hasil klasifikasi kualitas tomat buah?
2. Bagaimana unjuk kerja sistem klasifikasi kualitas tomat buah menggunakan metode *Grid Search SVM* berbasis *video processing*?

## **I.3 Tujuan Penelitian**

Tujuan akhir dari penelitian ini yaitu:

1. Untuk mengetahui pengaruh parameter  $C$  pada metode *Grid Search SVM* terhadap klasifikasi kualitas tomat buah.
2. Untuk mengetahui unjuk kerja sistem klasifikasi kualitas tomat buah menggunakan metode *Grid Search SVM* berbasis *video processing*.

## **I.4 Manfaat Penelitian**

Dengan dilakukannya penelitian ini, diharapkan manfaat yang didapatkan antara lain:

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



## **I.5 Batasan Masalah**

Yang menjadi batasan permasalahan dalam tugas akhir ini adalah:

1. Jenis tomat yang diuji adalah tomat buah.
2. Klasifikasi kualitas tomat buah dibagi menjadi 2 kategori yaitu layak dan tidak layak.
3. Sistem klasifikasi kualitas tomat buah menggunakan metode *Grid Search* SVM berbasis *video processing*.
4. Kamera yang digunakan yaitu *Logitech Webcam C922 Pro*.
5. Data sampel yang digunakan untuk menentukan kualitas tomat buah dari satu sisi yaitu bagian atas tomat buah.
6. Pengambilan data dilakukan dengan pencahayaan yang telah ditentukan.

## **I.6 Sistematika Penulisan**

Untuk memberikan gambaran singkat mengenai isi tulisan secara keseluruhan, maka akan diuraikan beberapa tahapan dari penulisan secara sistematis, yaitu:

### **BAB I PENDAHULUAN**

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

### **BAB II TINJAUAN PUSTAKA**

Bab ini berisi teori-teori terkait hal-hal yang mendasari dan yang berhubungan dengan penelitian, termasuk di dalamnya tomat,

pengolahan citra, visi komputer dan metode-metode yang digunakan dalam penelitian.

### **BAB III METODOLOGI PENELITIAN**

Bab ini memaparkan tahap penelitian, waktu dan lokasi penelitian, instrumen penelitian, teknik pengambilan data dan rancangan sistem serta penerapan algoritma *Grid Search SVM* dan teknik pengolahan data.

### **BAB IV HASIL DAN PEMBAHASAN**

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

### **BAB V PENUTUP**

Pada bab ini berisi kesimpulan yang diperoleh dari hasil penelitian yang dilakukan serta saran-saran untuk pengembangan sistem yang lebih lanjut.

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Tomat**

Tomat (*Lycopersicon esculentum*) merupakan tumbuhan asli Amerika Tengah dan Selatan dari keluarga *Solanaceae*. Kata “tomat” berasal dari kata dalam bahasa Nahuatl, dimana tomat merupakan keluarga dekat dari kentang. Tomat merupakan tumbuhan siklus hidup singkat, dapat tumbuh setinggi sekitar 1 sampai 3 meter. Tomat sering dianggap sebagai sayuran meskipun strukturnya adalah struktur buah. Tomat banyak dimanfaatkan sebagai sayuran, bumbu masak, buah meja, bahan pewarna, dan kosmetik. Tomat mengandung zat antioksidan cukup tinggi yang membantu untuk meningkatkan kekebalan tubuh, menghaluskan dan mencerahkan kulit, mencegah hipertensi, dan lainnya (Shafira, 2018).

Tomat merupakan komoditas unggulan hortikultura dengan nilai penting di Indonesia. Di Indonesia, kebutuhan pasar sayuran terutama buah tomat dari tahun ke tahun mengalami peningkatan yang dicerminkan dari angka produksi tomat. Kendala yang sering dihadapi yaitu kurang tersedianya tomat varietas unggul yang mempunyai produksi tinggi, buah berkualitas baik, dan tahan terhadap hama dan penyakit. Tingginya kebutuhan akan tomat, sehingga sangat penting untuk melakukan pemilihan atau sortasi tomat dengan kualitas baik untuk memenuhi kebutuhan dalam maupun luar negeri yang dalam hal ini sering kali terjadi ketidaksesuaian antara kualitas yang diperlukan dengan kualitas produk yang dihasilkan (Shafira, 2018).

## 2.2 Pengolahan Citra

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).

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.  $\sigma$  adalah nilai *standard deviation* distribusi normal yang digunakan. Makin besar nilai  $\sigma$ , 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.  $\pi$  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, & Faticah, 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:

$\mu$  = Mean

$M \times N$  = Ukuran Citra

$P$  = 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).

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

Keterangan:

$\sigma$  = *Standard deviation*

$M \times N$  = Ukuran Citra

$X$  = *Pixel Citra*

$\bar{X}$  = *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 ukurn 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 perhitungan terhadap nilai-nilai RGB itu sendiri dengan menggunakan rumus seperti berikut :

$$H = \tan^{-1} \left( \frac{\sqrt{3}(G-B)}{(R-G)+(R-B)} \right) \quad (2.5)$$

$$S = 1 - \min(R,G,B) / V \quad (2.6)$$

$$V = (R+G+B) / 3 \quad (2.7)$$

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 :

$$r = \frac{R}{R+G+B} \quad g = \frac{G}{R+G+B} \quad b = \frac{B}{R+G+B} \quad (2.8)$$

$$V = \max(r, g, b)$$

$$S = \begin{cases} 0, & \text{jika } V = 0 \\ 1 - \frac{\min(r, g, b)}{V}, & \text{jika } V > 0 \end{cases} \quad (2.9)$$

$$H = \begin{cases} 0, & \text{jika } S = 0 \\ \frac{60 \cdot (g-b)}{S \cdot V}, & \text{jika } V = r \\ 60 \cdot \left[ 2 + \frac{b-r}{S \cdot V} \right], & \text{jika } V = g \\ 60 \cdot \left[ 4 + \frac{r-g}{S \cdot V} \right], & \text{jika } V = b \end{cases} \quad (2.10)$$

$$H = H + 360 \text{ jika } H < 0 \quad (2.11)$$

Keterangan :

$H = Hue$  (warna sebenarnya)

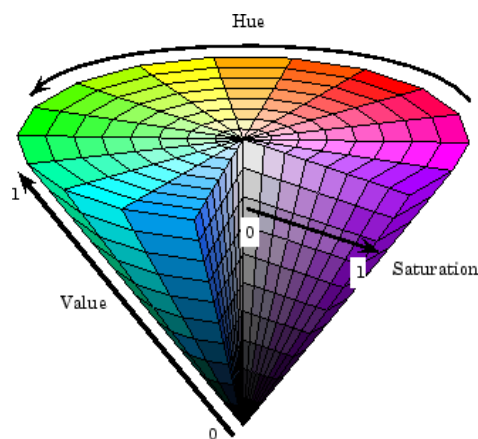
$S = Saturation$  (kemurnian warna)

$V = Value$  (kecerahan)

$R/r = red$  (merah)

$G/g = green$  (hijau)

$B/b = blue$  (biru)



**Gambar 2.1.** 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,  $\alpha_1$  atau  $\alpha_2$  (Zarwani, 2018). Rumus untuk menentukan nilai *threshold* bias didapatkan dari persamaan sebagai berikut:

$$g(x, y) = \begin{cases} 1 & \text{if } f(x,y) \geq T \\ 0 & \text{if } f(x,y) \leq T \end{cases} (x,y) \quad (2.12)$$

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).

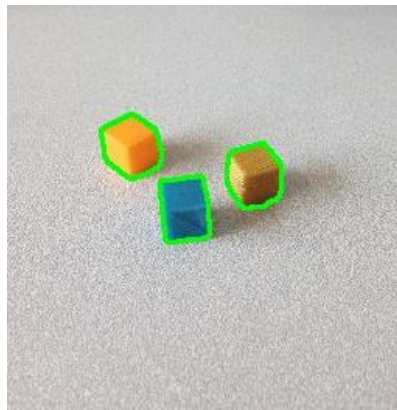
## 2.8 Blob Detection

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.



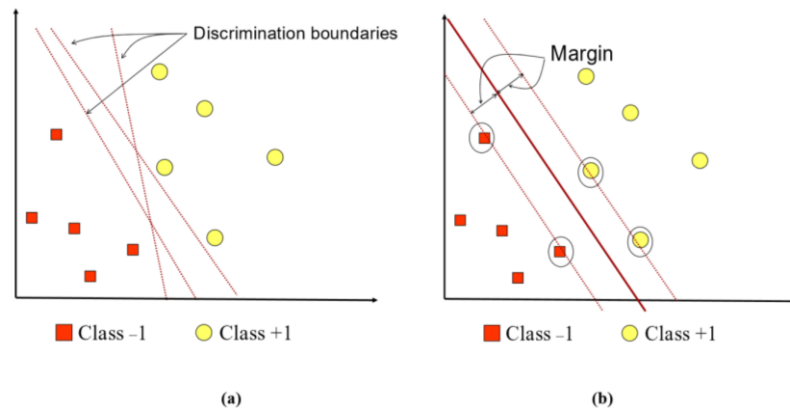
**Gambar 2.2.** Area Blob

(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 2.3.** Ilustrasi Pencarian *Hyperplane* Terbaik Menggunakan SVM

Gambar 2.3 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  $y_i = \pm 1$  yaitu kelas positif (+1) dan kelas negatif(-1) sehingga akan diperoleh *decision function* berikut.

$$f(x) = \text{sign}(w \cdot x + b) \quad (2.13)$$

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.

$$\min \tau(w) = \frac{1}{2} \|w\|^2 \quad (2.14)$$

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

$$y_i (w \cdot x_i + b) \geq 1 \quad \forall i \quad (2.15)$$

Persamaan di atas merupakan permasalahan optimisasi kendala dimana kita meminimalkan fungsi objek pada persamaan (2.14) dengan kendala pada persamaan (2.15). Permasalahan di atas dapat direduksi dengan menggunakan fungsi *Lagrange* yang terdiri dari jumlahan fungsi objektif dan m kendala dikalikan dengan pengganda *Lagrange* seperti berikut.

$$L(w, b) = \frac{1}{2} (w \cdot w) - \sum_{i=1}^m \alpha_i (y_i(w \cdot x_i + b) - 1) \quad (2.16)$$

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.

$$\frac{\partial L}{\partial b} = - \sum_{i=1}^m \alpha_i y_i = 0 \quad (2.17)$$

$$\frac{\partial L}{\partial w} = - \sum_{i=1}^m \alpha_i y_i x_i = 0 \quad (2.18)$$

Substitusi nilai w dari persamaan (2.18) ke dalam bentuk  $L(w,b)$  sehingga akan diperoleh rumus ganda atau biasa disebut sebagai *wolfe dual*.

$$W(\alpha) = \sum_{i=1}^m \alpha_i - \sum_{i=1}^m \alpha_i \alpha_j y_i y_j (x_i x_j) = 0 \quad (2.19)$$

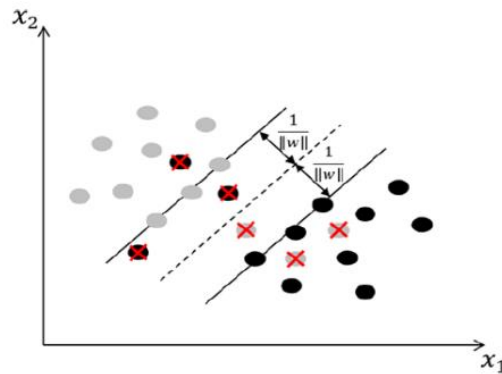
Dimana nilai  $\alpha_i$  terhadap kendala adalah sebagai berikut.

$$\alpha_i \geq 0 \quad \sum_{i=1}^m \alpha_i y_i = 0 \quad (2.20)$$

- *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).



(Sumber : Buku *Efficient Learning Machine*, 2015)

**Gambar 2.4.** Beberapa Misklasifikasi Pada *Soft Margin* SVM

Berdasarkan pada Gambar 3.3 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 (2.16) sebelumnya seperti berikut.

$$y_i (w \cdot x_i + b) \geq 1 - \varepsilon_i \quad (2.21)$$

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

$$\min \left[ \frac{1}{2} w \cdot w + c \sum_{i=1}^m \varepsilon_i [y_i (w \cdot x_i + b) - 1 + \varepsilon_i] - \sum_{i=1}^m r_i \varepsilon_i \right] \quad (2.22)$$

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

$$\min \tau(w) = \frac{1}{2} \|w\|^2 + c \sum_{i=1}^m \varepsilon_i \quad (2.23)$$

Parameter  $C$  digunakan untuk mengontrol *trade off* antara *margin* dan kesalahan klasifikasi  $\varepsilon$ .  $C$  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) :

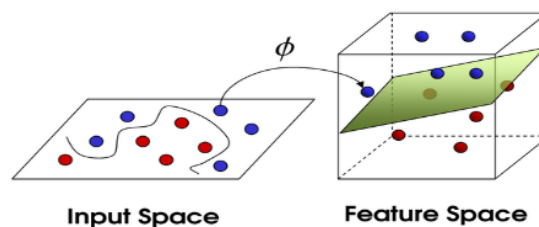
1. Secara prinsip SVM adalah *linear classifier*
2. *Pattern recognition* dilakukan dengan mentransformasikan data *input space* ke ruang yang berdimensi lebih tinggi, dan optimisasi dilakukan pada ruang vektor yang baru tersebut. Hal ini membedakan SVM dari solusi *pattern recognition* pada umumnya, 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 dikostuksi 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*”.



(Sumber : [www.quora.com](http://www.quora.com))

**Gambar 2.5.** Kernel SVM Untuk Memisahkan Data Secara *Linear*



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.

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

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

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

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

$$\mathbf{K}(\mathbf{x}, \mathbf{z}) = \Phi(\mathbf{x})^T \Phi(\mathbf{z}) \tag{2.24}$$

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

$$\Phi(\mathbf{x}) = \{ x_1^2, \sqrt{2}x_1 x_2, x_2^2 \} \tag{2.25}$$

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

$$\Phi: X \rightarrow F$$

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

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

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

$$f(\Phi(x)) = \text{sign}(\sum_{i=1}^m a_i y_i K(x,z) + b) \quad (2.26)$$

dimana :

$b$  : Nilai bias

$m$  : Jumlah *support vector*

$K(x,z)$  : Fungsi nilai kernel

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

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

$$\int dx \int dz f(x) K(x,z) f(z) > 0 \quad (\forall f \in L_2) \quad (2.27)$$

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

$$K(x,z) = K_1(x,z) + K_2(x,z): \text{Direct sum} \quad (2.28)$$

$$K(\mathbf{x}, \mathbf{z}) = \alpha K_1(\mathbf{x}, \mathbf{z}) \quad : \text{Skalar product} \quad (2.29)$$

$$K(\mathbf{x}, \mathbf{z}) = K_1(\mathbf{x}, \mathbf{z}) K_2(\mathbf{x}, \mathbf{z}) \quad : \text{Direct product} \quad (2.30)$$

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.

$$K(\mathbf{x}, \mathbf{z}) = \mathbf{x}^T \mathbf{z} \quad (2.31)$$

Pemetaan fungsi  $\Phi$  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.

$$K(\mathbf{x}, \mathbf{z}) = (\mathbf{x}^T \mathbf{z})^d \text{ atau } (1 + \mathbf{x}^T \mathbf{z})^d \quad (2.32)$$

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.

$$K(\mathbf{x}, \mathbf{z}) = \exp[-\gamma \|\mathbf{x} - \mathbf{z}\|^2] \quad (2.33)$$

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 *Cross Validation and Grid Search SVM*

*Cross validation* adalah metode statistik untuk mengevaluasi dan membandingkan algoritma pembelajaran dengan membagi data menjadi dua bagian yaitu data latih dan dataset. Semua data yang dikelompokkan ke dalam dua bagian tersebut akan secara bergantian digilir ke dalam bagian lainnya secara berurutan (Refailzadeh, et, al., 2008).

Metode *Grid Search SVM* merupakan salah satu alternatif untuk mencari parameter terbaik untuk suatu model dengan memberi *range* nilai parameter tersebut, sehingga pengklasifikasi dapat secara akurat memprediksi data yang tidak berlabel (yaitu data pengujian). Metode tersebut dikategorikan sebagai metode lengkap karena nilai parameter terbaik harus dieksplorasi masing-masing dengan menetapkan jenis nilai prediksi terlebih dahulu. Kemudian metode akan menampilkan skor untuk setiap parameter nilai untuk dipertimbangkan mana yang akan dipilih. Metode ini dapat diterapkan jika maksimum yang diperlukan diketahui berada dalam area terbatas yang ditentukan oleh batas atas dan bawah dari masing-masing variabel independen (Lameski, Zdravevski, Mingov, & Kulakov, 2015).

Algoritma *Grid Search SVM* ini biasanya menggunakan fungsi *k-fold cross validation*. Pencarian parameter terbaik akan dilakukan dengan cara membagi data menggunakan *k-fold cross validation* (Rosidah, 2018). Dalam *k-fold cross validation* dataset yang utuh akan dipecah secara random menjadi 'k' subset dengan ukuran yang sama dan saling eksklusif satu dengan yang lainnya. Setiap kali pelatihan semua akan dilatih pada semua *fold* kecuali hanya satu *fold* yang disisakan

untuk pengujian. Namun, pada dasarnya jika nilai *cv* dalam *grid search* tidak ada maka untuk menggunakan validasi silang 5 kali lipat secara *default*.

## **2.11 Penelitian Terkait**

### **2.11.1 Implementasi *Convolutional Neural Networks* Untuk Klasifikasi Citra Tomat Menggunakan Keras (Shafira, 2018)**

Penelitian ini menggunakan metode *Convolutional Neural Networks* (*CNN*) dikarenakan ke dalaman jaringan yang tinggi dan banyak diaplikasikan pada data citra. Jaringan pada *CNN* mempunyai lapisan khusus yang disebut dengan lapisan konvolusi. Proses konvolusi citra pada penelitian ini menggunakan package Keras pada *software* RStudio versi 1.1.383, dikarenakan pembuatan model jaringan syaraf menggunakan Keras tidak perlu menuliskan kode untuk mengekspresikan perhitungan matematisnya satu persatu. Pengujian dengan sampel 100 citra tomat dengan 3 macam skenario untuk mengklasifikasikan tomat dalam 2 kelas yaitu tomat layak dan tidak layak. Tingkat akurasi sebesar 90% yang didapatkan dari skenario yang ke 3, dengan perbandingan data training dan data testing sebesar 80%:20% yang dinilai telah mampu melakukan identifikasi kelayakan buah tomat.

### **2.11.2 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%.

### **2.11.3 *The Development of Machine Vision System for Sorting Passion Fruit using MultiClass Support Vector Machine (Sidehabi, Suyuti, Areni, & Nurtanio, 2018)***

Sitti Wetenriajeng melakukan penelitian mengenai pengembangan Sistem *Machine Vision* untuk Sortasi Buah Markisa yang bertujuan untuk mengembangkan proses pemilahan buah markisa secara otomatis berdasarkan tingkat kematangannya dengan menggunakan *webcam* yang dikombinasikan dengan aplikasi *computer vision* untuk melakukan tugas pemilahan buah markisa. Metode *computer vision* yang diusulkan adalah algoritma *K-Means Cluster* sebagai ekstraksi ciri, dengan *Multi Class Support Vector Machine* dengan kernel dengan kernel *linear*, RBF, *Sigmoid* dan *Polynomial* untuk pemilahan tingkat kematangan buah markisa dalam tiga kategori yaitu matang, hampir matang dan mentah dengan input data berupa 6 sisi buah markisa dengan jenis data yang digunakan berupa video sedangkan untuk optimasi parameter pada fungsi kernel dilakukan dengan metode *grid-search*. Sehingga mendapatkan tingkat akurasi sebesar 93.3% dengan metode SVM kernel RBF.

#### **2.11.4 A Deep Learning Method for Recognizing Elevated Mature Strawberries (Li, Li, & Tang, 2018)**

Li, dkk melakukan penelitian untuk mengenali stroberi yang matang langsung pada pohonnya. Sistem yang dibuat menggunakan metode *deep learning*. Sistem ini untuk membantu mesin dalam memisahkan buah stroberi dari daun dan stroberi yang saling tumpang tindih. Penelitian ini menggunakan metode *otsu* untuk tahap segmentasi, HOG dan H component untuk ekstraksi fitur sedangkan untuk tahap pengenalan buah stroberi yang matang menggunakan dua metode untuk dibandingkan yaitu *CaffeNet* yang merupakan salah satu arsitektur dari CNN (*Convolutional Neural Network*) dan SVM. Hasil menunjukkan bahwa pengenalan stroberi matang menggunakan metode *CaffeNet* memiliki akurasi lebih baik 11% dari SVM yaitu akurasi masing-masing 95% dan 84%.

#### **2.11.5 Using machine learning techniques for evaluating tomato ripeness. (El-Bendary, El Hariri, Hassanien, & Badr, 2015)**

Penelitian ini menggunakan *One-against-one*(OAO) *multi class* SVM dengan kernel *linear*, *One-against-all*(OAA) *multi class* SVM dengan kernel *linear* dan *Linear Discriminant Analysis*(LDA) yang digunakan untuk ekstraksi dan klasifikasi fitur. Pendekatan ini bertujuan untuk pengukuran dan evaluasi kematangan tomat melalui penyelidikan dan klasifikasi tahap kematangan yang berbeda dengan jenis inputan data berupa gambar. Penelitian telah memperoleh akurasi klasifikasi kematangan sebesar 90,80%, menggunakan algoritma *multi-*



*class* SVM satu lawan satu (OAO) dengan fungsi kernel *linear*, akurasi klasifikasi kematangan 84,80% menggunakan algoritme *multi-kelas* SVM satu lawan semua (OAA) dengan kernel *linear* fungsi, dan akurasi klasifikasi kematangan 84% menggunakan algoritma LDA.