

**DETEKSI KEMATANGAN BUAH NANAS BERDASARKAN TINGKAT  
KEKUNINGAN MATA NANAS DENGAN MENGGUNAKAN METODE  
*TRESHOLDING***

(Skripsi)

Oleh

**Wayan Putri Yanti**



**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS LAMPUNG  
BANDAR LAMPUNG  
2022**

## ABSTRAK

### DETEKSI KEMATANGAN BUAH NANAS BERDASARKAN TINGKAT KEKUNINGAN MATA NANAS DENGAN MENGGUNAKAN METODE *TRESHOLDING*

Oleh

Wayan Putri Yanti

Telah dilakukan penelitian deteksi kematangan buah nanas berdasarkan tingkat kekuningan mata nanas dengan menggunakan metode *thresholding*. Tujuan dari penelitian ini adalah menghasilkan alat deteksi yang dapat menentukan tingkat kematangan buah nanas berdasarkan persentase kematangan buah nanas dengan metode *thresholding*. Metode ini dilakukan untuk memisahkan mata nanas dengan *background*-nya. Pada penelitian ini dilakukan dua langkah *thresholding* yaitu mencari jumlah seluruh mata nanas dengan format gambar RGB dan mencari jumlah mata nanas yang kuning dengan format gambar HSV. Selanjutnya dilakukan perhitungan untuk persentase kematangan buah nanas dengan rumus jumlah mata kuning dibagi jumlah seluruh mata nanas dikali seratus persen. Hasil yang diperoleh mempunyai akurasi rata-rata kematangan sebesar  $(93 \pm 2)\%$ .

**Kata kunci:** *Computer vision*, nanas, *thresholding*.

## **ABSTRACT**

### ***DETECTION OF PINEAPPLE RIPENESS BASED ON THE YELLOWISH LEVEL OF PINEAPPLE EYES USING THE THRESHOLDING METHOD***

**By**

**Wayan Putri Yanti**

*A research on detection of pineapple ripeness has been carried out based on the yellowish level of pineapple eyes by using the thresholding method. The purpose of this study is to produce a detection tool that can determine the maturity level of pineapple based on the percentage of pineapple ripeness using the thresholding method. This method is used to separate the pineapple eyes from the background. In this study, two thresholding steps were carried out, namely looking for the total number of pineapple eyes with RGB image format and looking for the number of yellow pineapple eyes with HSV image format. Furthermore, the calculation for the percentage of pineapple ripeness is carried out with the formula for the number of yellow eyes divided by the total number of pineapple eyes multiplied by one hundred percent. The results obtained have an average maturity accuracy of  $(93 \pm 2)\%$ .*

**Keywords:** *Computer vision, pineapple, thresholding.*

**DETEKSI KEMATANGAN BUAH NANAS BERDASARKAN TINGKAT  
KEKUNINGAN MATA NANAS DENGAN MENGGUNAKAN METODE  
*TRESHOLDING***

**Oleh**

**Wayan Putri Yanti**

**Skripsi**

**Sebagai Salah Satu Syarat untuk Mencapai Gelar  
SARJANA SAINS**

**Pada**

**Jurusan Fisika  
Fakultas Matematika dan Ilmu Pengetahuan Alam**



**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS LAMPUNG  
BANDAR LAMPUNG  
2022**

Judul Skripsi : **DETEKSI KEMATANGAN BUAH NANAS  
BERDASARKAN TINGKAT KEKUNINGAN  
MATA NANAS DENGAN MENGGUNAKAN  
METODE *TRESHOLDING***

Nama Mahasiswa : **Wayan Putri Yanti**

Nomor Pokok Mahasiswa : **1817041053**

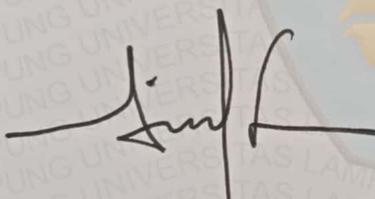
Jurusan : **Fisika**

Bidang Keahlian : **Instrumentasi**

Fakultas : **Matematika dan Ilmu Pengetahuan Alam**

**MENYETUJUI**

1. **Komisi Pembimbing**

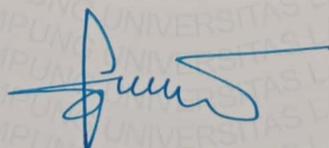


**Arif Surtono, S.Si., M.Si., M.Eng.**  
NIP. 197109092000121001



**Donni Kis Apriyanto, S.Si., M.Sc.**  
NIP. 198804032019031005

2. **Ketua Jurusan Fisika**

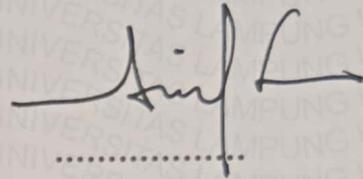


**Gurum Ahmad Pauzi, S.Si., M.T.**  
NIP. 198010102005011002

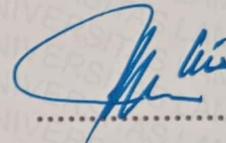
**MENGESAHKAN**

1. Tim Penguji

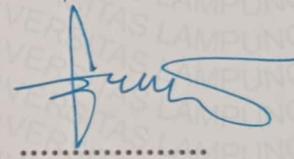
Ketua : **Arif Surtono, S.Si., M.Si., M.Eng.**



Sekretaris : **Donni Kis Apriyanto, S.Si., M.Sc.**



Penguji  
Bukan Pembimbing : **Gurum Ahmad Pauzi, S.Si., M.T.**



2. Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam



**Dr. Eng. Satripto Dwi Yuwono, S.Si., M.T.**  
NIP. 197407052000031001

Tanggal Lulus Ujian Skripsi: **24 Novembar 2022**

## PERNYATAAN

Dengan ini saya menyatakan bahwa dalam skripsi ini tidak terdapat karya yang pernah dilakukan orang lain dan sepengetahuan saya tidak ada karya atau pendapat yang ditulis atau diterbitkan oleh orang lain kecuali yang secara tertulis diacu dalam naskah ini sebagaimana disebutkan dalam daftar pustaka. Selain itu, saya menyatakan pula bahwa skripsi ini dibuat oleh saya sendiri.

Apabila pernyataan saya tidak benar maka saya bersedia dikenai sanksi sesuai dengan hukum yang berlaku.



Bandar Lampung, 1 Desember 2022

Wayan Putri Yanti  
NPM. 1817041053

## RIWAYAT HIDUP



Penulis bernama lengkap Wayan Putri Yanti yang lahir pada tanggal 15 Februari 2000 di Rejo Binangun, dari pasangan Bapak Wayan Nada Sukarya dan Ibu Made Rusni sebagai anak pertama dari dua bersaudara. Penulis menyelesaikan pendidikan di TK PKK Bali Indah pada tahun 2006, SDN 2 Rejo Binangun pada tahun 2012, SMPN 1 Raman Utara pada tahun 2015, SMAN 1 Raman Utara pada tahun 2018. Pada tahun 2018, penulis diterima sebagai mahasiswa di Universitas Lampung, Jurusan Fisika, Fakultas Matematika dan Ilmu Pengetahuan Alam melalui jalur Seleksi Bersama Masuk Perguruan Tinggi Negeri (SBMPTN).

Selama menjadi mahasiswa, penulis aktif dalam Himpunan Mahasiswa Fisika (HIMAFI) sebagai anggota biro Dana dan Usaha (Danus) serta UKM Hindu Universitas Lampung sebagai anggota bidang Penelitian dan Pengembangan (Litbang). Penulis juga pernah mengikuti program Kampus Merdeka yaitu Pertukaran Mahasiswa Merdeka (PMM) di Universitas Negeri Semarang sebagai kampus tujuan dan Universitas Gadjah Mada sebagai kampus mitra. Selain PMM, peneliti juga mengikuti program Kampus Merdeka yang lain yaitu Study Independen di PT. Bisa AI. Kemudian penulis pernah menjadi asisten Praktikum Elektronika Dasar 2. Penulis melaksanakan Praktik Kerja Lapangan (PKL) di PT. PLN Unit Induk Distribusi Lampung dengan judul **“Pengaruh Tegangan Kirim**

**Trafo Distribusi Terhadap Susut Jaringan Tegangan Rendah dengan Arus 140 A ”.** Selain itu, Penulis melaksanakan Kuliah Kerja Nyata (KKN) di Rejo Binangun, Kecamatan Raman Utara, Kabupaten Lampung Timur pada tanggal 1 Februari – 10 Maret 2021.

## MOTTO

Hidup itu adalah Sebuah Perjalanan yang Penuh Perjuangan

“Salah Satu Cara Melakukan Pekerjaan yang Hebat adalah dengan Mencintai  
Apa Yang Kamu Lakukan”  
~**Steve Jobs**~

“Bermimpilah Dalam Hidup, Jangan Hidup dalam Mimpi”  
~**Andrea Hirata**~

“Anda Mungkin Bisa Menunda, Tapi Waktu Tidak Akan Menunggu”  
~**Benjamin Franklin**~

## **PERSEMBAHAN**

Dengan rasa syukur kepada Ida Sang Hyang Widhi Wasa

Saya persembahkan karya ini kepada:

### **Kedua Orang Tua Tercinta**

#### **Bapak Wayan Nada Sukarya dan Ibu Made Rusni**

Terimakasih doa dan semangat kalian yang tanpa lelah membangun kepercayaan diriku sehingga aku dapat menempuh semua masalah dengan semangat dan pantang menyerah

### **Bapak Ibu Guru serta Bapak Ibu Dosen**

Terimakasih atas segala bimbingan dan didikan yang telah diberikan, semoga ilmu yang kalian berikan dapat bermanfaat dan menjadi bekal untuk kesuksesan saya kelak

### **Adik tercinta, Made Jaya Setiawan**

Terimakasih atas segala dukungan dan semangatnya, dan selalu ada saat aku membutuhkan sandaran

**Rekan-rekan Fisika FMIPA Unila angkatan 2018**

Almamater Tercinta

**“UNIVERSITAS LAMPUNG”**

## KATA PENGANTAR

Puji dan syukur penulis panjatkan kepada Tuhan Yang Maha Esa, yang telah melimpahkan kesehatan dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul “**Deteksi Kematangan Buah Nanas Berdasarkan Tingkat Kekuningan Mata Nanas Dengan Menggunakan Metode Thresholding**”. Tujuan penulisan skripsi ini adalah sebagai salah satu persyaratan untuk mendapatkan gelar SI dan untuk melatih mahasiswa bagaimana cara berpikir cerdas, kreatif dan inovatif dalam menulis karya ilmiah.

Penulis menyadari terdapat banyak kekurangan dalam skripsi ini. Oleh karena itu, saran dan kritik yang membangun sangat penulis harapkan agar skripsi ini menjadi lebih baik. Akhir kata, semoga skripsi ini dapat bermanfaat bagi semua.

Bandar Lampung, 24 November 2022

Penulis,

**Wayan Putri Yanti**

## SANWACANA

Puji dan syukur penulis ucapkan kepada Tuhan Yang Maha Esa, karena atas karunia-Nya penulis dapat mengucapkan ungkapan terimakasih kepada pihak-pihak yang telah membantu penulis dalam menyelesaikan penelitian dan skripsi ini, terutama kepada:

1. Bapak Arif Surtono, S.Si., M.Si., M.Eng., selaku pembimbing I yang telah membimbing dengan sabar dan memberikan motivasi serta arahan yang mendukung dari awal hingga akhir penulisan.
2. Bapak Donni Kis Apriyanto, S.Si., M.Sc., selaku pembimbing II yang dengan telaten membimbing perancangan program deteksi dan memberikan masukan-masukan yang membangun untuk menyelesaikan skripsi dari awal hingga akhir penulisan.
3. Bapak Gurum Ahmad Pauzi, S.Si., M.T., selaku penguji dan Ketua Jurusan yang telah memberikan bimbingan, serta mengoreksi kekurangan, dan memberikan kritik dan saran yang membangun dari awal hingga akhir penulisan.
4. Bapak Dr. rer. nat. Roniyus Marjunus, M.Si., selaku Pembimbing Akademik yang telah memberikan arahan dan motivasi selama perkuliahan.
5. Bapak Dr. Eng. Sripto Dwi Yuwono, S.Si., M.T., selaku Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam.

6. Segenap dosen yang telah memberikan ilmu pengetahuan yang bermanfaat bagi penulis selama menempuh pendidikan di Jurusan Fisika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Lampung.
7. Kedua orang tuaku yang selalu menyemangatiku agar selalu semangat dan jangan gampang menyerah sehingga penulis memiliki semangat untuk menyelesaikan skripsi ini.
8. Aish, Intan, Caca, Ester, dan Elda yang selalu menemani dan memberikan dukungan dalam menyelesaikan tugas akhir ini.

Semoga Tuhan Yang Maha Esa memberikan nikmatnya kepada kita semua.

Bandar Lampung, 24 November 2022

Penulis

Wayan Putri Yanti

## DAFTAR ISI

	<b>Halaman</b>
<b>ABSTRAK</b> .....	<b>i</b>
<b>ABSTRACT</b> .....	<b>ii</b>
<b>HALAMAN JUDUL</b> .....	<b>iii</b>
<b>LEMBAR PERSETUJUAN</b> .....	<b>iv</b>
<b>LEMBAR PENGESAHAN</b> .....	<b>v</b>
<b>HALAMAN PERNYATAAN</b> .....	<b>vi</b>
<b>RIWAYAT HIDUP</b> .....	<b>vii</b>
<b>MOTTO</b> .....	<b>ix</b>
<b>PERSEMBAHAN</b> .....	<b>x</b>
<b>KATA PENGANTAR</b> .....	<b>xi</b>
<b>SANWACANA</b> .....	<b>xii</b>
<b>DAFTAR ISI</b> .....	<b>xiv</b>
<b>DAFTAR GAMBAR</b> .....	<b>xvi</b>
<b>DAFTAR TABEL</b> .....	<b>xvii</b>

### **I. PENDAHULUAN**

1.1. Latar Belakang .....	1
1.2. Rumusan Masalah .....	4
1.3. Tujuan Penelitian.....	4
1.4. Manfaat Penelitian.....	4
1.5. Batasan Masalah .....	5

### **II. TINJAUAN PUSTAKA**

2.1. Penelitian Terkait .....	6
2.2. Teori Dasar .....	11
2.2.1. Nanas .....	11
2.2.2. Pengolahan Citra .....	14
2.2.3. RGB .....	16
2.2.4. HSV .....	18

2.2.5. <i>Thresholding</i> .....	20
2.2.6. Morfologi Citra .....	22
2.2.7. Operasi <i>Bitwise</i> dan <i>Masking</i> .....	25
2.2.8. Kontur .....	27
2.2.9. <i>Python</i> dan <i>OpenCV</i> .....	29
2.2.10. Webcam.....	33

### III. METODE PENELITIAN

3.1. Tempat dan Waktu Penelitian .....	35
3.2. Alat dan Bahan .....	35
3.3. Prosedur Penelitian.....	36
3.3.1. Studi Literatur.....	37
3.3.2. Perancangan Sistem Deteksi.....	37
3.3.3. Menghitung Jumlah Mata Nanas.....	39
3.3.3.1. Menghitung Jumlah Seluruh Mata Nanas.....	39
3.3.3.2. Menghitung Jumlah Mata Nanas yang Kuning.....	42
3.3.4. Klasifikasi Kematangan Nanas.....	45
3.3.5. Pengujian dan Analisis Sistem.....	48

### IV. HASIL DAN PEMBAHASAN

4.1. Implementasi Perancangan Hardware Sistem Deteksi Kematangan Nanas.....	49
4.2. Implementasi Software Sistem Deteksi Kematangan Nanas.....	50
4.2.1. Hasil Perancangan <i>Graphical User Interface</i> (GUI) .....	51
4.2.2. Kontrol <i>Graphical User Interface</i> (GUI) .....	52
4.2.2.1. Fungsi Kendali <i>Pushbutton</i> .....	53
4.2.2.2. Fungsi Program Untuk Memulai ( <i>Start</i> ) .....	54
4.2.2.3. Fungsi Program Untuk Mengambil Gambar Nanas .....	55
4.2.2.4. Fungsi Program Untuk Mendeteksi Kematangan Nanas .....	57
4.2.2.5. Fungsi Program Untuk Menyimpan Gambar .....	62
4.2.2.6. Fungsi Program Untuk Keluar Dari GUI .....	64
4.3. Hasil Pengujian Sistem.....	65
4.3.1. Hasil Pengujian Akuisisi Data .....	65
4.3.2. Analisis Persentase Kematangan Nanas .....	69
4.3.3. Analisis Kestabilan Alat Deteksi Kematangan Nanas.....	72

### V. KESIMPULAN

5.1. Kesimpulan.....	75
5.2. Saran.....	75

### DAFTAR PUSTAKA

### LAMPIRAN

## DAFTAR GAMBAR

Gambar	Halaman
2.1. Perangkat pendeteksi kematangan nanas.....	6
2.2. Purwarupa pemilah nanas .....	8
2.3. Ilustrasi secara real-time dari setup eksperimental di perkebunan nanas ...	8
2.4. (a) Gambar hasil ekstraksi; (b) penghapusan latar belakang menggunakan ruang warna HSV; (c) citra biner; (d) operasi morfologi; (e) deteksi mahkota kotak pembatas nanas .....	9
2.5. (a) Gambar hasil ekstraksi; (b) penghapusan latar belakang menggunakan ruang warna $L^*A^*B$ ; (c) citra biner; (d) operasi morfologi; (e) deteksi mahkota kotak pembatas nanas .....	10
2.6. Tingkat kematangan buah nanas.....	13
2.7. Koordinat citra digital.....	15
2.8. Citra RGB .....	17
2.9. (a) Ruang warna HSV (b) Representasi HSV .....	19
2.10. RGB ke HSV.....	20
2.11. Metode <i>Thresholding</i> mengubah citra <i>grayscale</i> menjadi biner.....	22
2.12. (a) citra <i>grayscale</i> (b) citra biner .....	22
2.13. (a) citra asli, (b) struktur kernel, (c) citra hasil dilasi.....	23
2.14. (a) citra asli, (b) struktur kernel, (c) citra hasil erosi .....	23
2.15. (a) citra asli, (b) struktur kernel, (c) citra hasil opening.....	24
2.16. (a) citra asli, (b) struktur kernel, (c) citra hasil closing .....	25
2.17. Proses Masking .....	26
2.18. Kontur .....	28
2.17. logo <i>python</i> .....	29
2.18. Interpreter.....	31
2.19. Logo <i>openCV</i> .....	32
2.20. Webcam Logitech C992 Pro HD Stream.....	34

3.1. Diagram Alir Penelitian.....	36
3.2. Diagram blok system identifikasi kematangan buah nanas .....	37
3.3. Diagram blok deteksi kematangan buah nanas.....	38
3.4. Desain <i>interface</i> deteksi kematangan buah nanas.....	38
3.5. Diagram Alir segmentasi menghitung jumlah seluruh jumlah mata nanas	39
3.6. Diagram Alir segmentasi menghitung jumlah mata nanas kuning.....	42
3.7. Diagram Alir Sistem.....	46
4.1. Alat Deteksi Kematangan Nanas dengan Metode <i>Thresholding</i> .....	49
4.2. Kotak Deteksi Kematangan Nanas .....	50
4.3. (a) Power Suply 12 V 5 A (b) Potensiometer.....	50
4.4. Tampilan <i>Graphical User Interface</i> (GUI) .....	51
4.5. Tampilan GUI <i>Pushbutton Start</i> .....	65
4.6. Tampilan GUI <i>Pushbutton Capture</i> .....	66
4.7. Tampilan GUI <i>Pushbutton Save</i> .....	66
4.8. Tampilan GUI Saat Gambar Berhasil Tersimpan .....	67
4.9. Tampilan GUI Hasil Pemrosesan Gambar Nanas .....	67
4.10. Tampilan GUI <i>Pushbutton Exit</i> .....	68
4.11. Grafik persentase kesalahan dan akurasi alat kematangan nanas .....	74

## DAFTAR TABEL

Tabel	Halaman
2.1. Representasi Hue .....	19
2.2. Operator Bitwise.....	25
3.1. Alat dan bahan penelitian.....	35
3.2. Rancangan tabel data klasifikasi kematangan nanas .....	45
3.3. Rancangan tabel data nilai kesalahan dan akurasi sistem deteksi.....	48
4.1. Mencari <i>threshold</i> untuk mencari jumlah seluruh mata nanas .....	58
4.2. Mencari <i>threshold</i> untuk jumlah mata kuning.....	61
4.3. Perbandingan Persentase Mata Nanas Referensi dan Hasil Deteksi .....	69
4.4. Kesalahan dan Akurasi Kematangan Buah Nanas .....	70
4.5. Kestabilan alat deteksi kematangan Nanas.....	72
4.6. Persentase kesalahan dan akurasi kestabilan alat sistem deteksi.....	73

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Buah nanas merupakan buah tropis yang cukup diminati masyarakat Indonesia karena memiliki rasa manis yang unik dan segar. Biasanya buah nanas dikonsumsi dalam bentuk buah segar, jus, atau buah dalam kaleng (Lubis, 2020). Kandungan air, gula, vitamin, dan mineral yang tinggi membuat buah nanas terasa manis dan segar serta kaya akan manfaat. Selain dikonsumsi, kulit buahnya juga dimanfaatkan untuk pakan ternak. Sedangkan serat pada daunnya dapat diolah menjadi kertas dan tekstil (Harahap dkk., 2019).

Salah satu industri yang membudidayakan buah nanas di Indonesia adalah PT. GGP (*Great Giant Pineapple*) yang terletak di Jl. Raya Arah Menggala Km 77 Kecamatan Terbanggi Besar Lampung Tengah. Perusahaan ini menggunakan teknologi dan cara bercocok tanam modern dengan luas lahan  $\pm$  32.000 ha (Zaman dkk., 2020). Produk buah yang dihasilkan dari perusahaan ini yaitu nanas, jambu, pepaya, dan pisang yang telah di ekspor ke lebih dari 60 negara di dunia. Buah nanas memberikan kontribusi terbesar dengan produksi sebesar 630 ribu ton per tahun (Suryadi, 2019). PT. GGP menghasilkan beberapa jenis produk nanas yaitu nanas dalam kaleng, jus, konsentrat nanas, dan nanas segar (Khoiriah, 2021).

Buah nanas yang dibudidayakan oleh PT. GGP adalah nanas varietas *Cayenne smooth* (Susanti, 2013). Setelah buah nanas panen maka dilakukan proses penyortiran. Namun, penyortiran nanas di PT. GGP masih dilakukan dengan cara manual, yaitu menggunakan penglihatan/tenaga manusia. Proses penyortiran seperti ini, tergantung kondisi dari kualitas penyortir seperti faktor kelelahan manusia, keragaman visual dan perbedaan persepsi mutu buah. Hal ini mengakibatkan hasil sortiran menjadi bersifat subjektif tergantung siapa yang menyortir. Oleh karena itu, perlu sentuhan teknologi dalam melakukan proses penyortiran buah nanas agar diperoleh mutu buah nanas yang memenuhi standar.

Beberapa metode telah dikembangkan oleh peneliti di dunia untuk mengklasifikasikan mutu buah nanas diantaranya teknik pengukuran dengan parameter non-destruktif, kimia, destruktif (Pathaveerat dkk., 2008), ekstraksi fitur warna buah (Asnor dkk., 2013; Bakar dkk., 2013; Ullah dkk., 2018), *Convolutional Neural Network* (CNN) (Azman dan Ismail, 2017), Segmentasi (Nawawi dan Ismail, 2017), deteksi Viola-Jones (Rahman, 2016), dan metode untuk menyesuaikan model kulit nanas dengan *Active Shape Model* (ASM) (Kaewapichai dkk., 2007). Namun metode-metode tersebut masih memiliki kelemahan. Pada metode destruktif, kimia, dan non-destruktif nanas harus dibelah agar dapat diidentifikasi sehingga dapat merusak buah nanas. Sedangkan untuk metode CNN dan Viola Jones membutuhkan banyak sampel objek dan latar belakang selama pelatihan sehingga membutuhkan banyak waktu untuk pelabelan yang dilakukan secara berulang-ulang. Berlaku juga untuk metode ASM, metode ini membutuhkan data set yang banyak untuk pemodelan nanas. Lalu pada metode ekstraksi warna dan segmentasi, warna kulit nanas dideteksi secara umum sehingga

warna dari setiap mata nanas tidak terdeteksi secara maksimal. Oleh karena itu, diperlukan metode baru untuk mengklasifikasi buah nanas sehingga dapat memperbaiki kinerja metode yang telah ada.

Metode deteksi kematangan buah nanas dengan cara menghitung persentase jumlah mata berwarna kuning (matang) merupakan metode baru yang diusulkan dalam penelitian ini. Dengan metode ini, persentase jumlah mata kuning terhadap jumlah total mata buah nanas lebih menggambarkan kematangan masing-masing buah nanas itu sendiri. Oleh karena itu kematangan buah nanas dihitung terhadap kematangan dirinya sendiri.

Penelitian ini diusulkan untuk memperbaiki alat pendeteksi kematangan nanas yang telah dilakukan sebelumnya. Metode yang akan digunakan adalah *thresholding* untuk mendeteksi tingkat kematangan buah nanas berdasarkan jumlah mata nanas yang kuning. Metode ini dapat mendeteksi tingkat kekuningan mata nanas pada sebuah nanas, sehingga akan mengurangi deteksi yang terlewatkan. Selain itu citra yang diambil langsung diproses dengan menggunakan metode *thresholding* sehingga gambar yang didapat langsung dapat diklasifikasikan kedalam standar tingkat kematangan nanas. Metode *thresholding* sendiri merupakan suatu metode yang dapat digunakan untuk memisahkan antara *obyek* dan *background*-nya. Langkah pertama yaitu menghitung jumlah seluruh mata nanas. Setelah menghitung jumlah mata nanas maka ditentukan titik tengah dari setiap mata nanas. Setiap mata nanas memiliki warna dengan rentang dari hijau kekuning. Dengan warna mata tersebut kita dapat mendeteksi warna kuning dengan HSV. Lalu digunakan dua nilai ambang dengan metode *thresholding global* sehingga

dihasilkan sebuah persentase tingkat kematangan buah nanas dengan perbandingan jumlah mata nanas yang kuning dan jumlah seluruh mata nanas.

## 1.2 Rumusan Masalah

Berdasarkan uraian pada latar belakang tersebut, maka dapat dirumuskan masalah sebagai berikut.

1. Bagaimana cara mendeteksi tingkat kematangan nanas dengan metode *thresholding*?
2. Bagaimana cara mendeteksi kematangan buah nanas menggunakan mata nanas sebagai objek deteksi?

## 1.3 Tujuan Penelitian

Tujuan dilakukan penelitian ini adalah sebagai berikut.

1. Membuat metode deteksi kematangan buah nanas berdasarkan tingkat persentase mata nanas berwarna kuning menggunakan teknik *thresholding*.
2. Membuat instrumentasi untuk mendeteksi kematangan buah nanas berdasarkan tingkat kekuningan dari mata nanas.

## 1.4 Manfaat Penelitian

Manfaat dari penelitian ini adalah sebagai berikut.

1. Dihasilkan *prototype* alat untuk mendeteksi tingkat kematangan buah nanas dengan penilaian persentase jumlah mata kuning (matang).
2. Dapat mendeteksi kematangan buah nanas secara akurat dan efisien berdasarkan jumlah mata nanas yang kuning.

## 1.5 Batasan Masalah

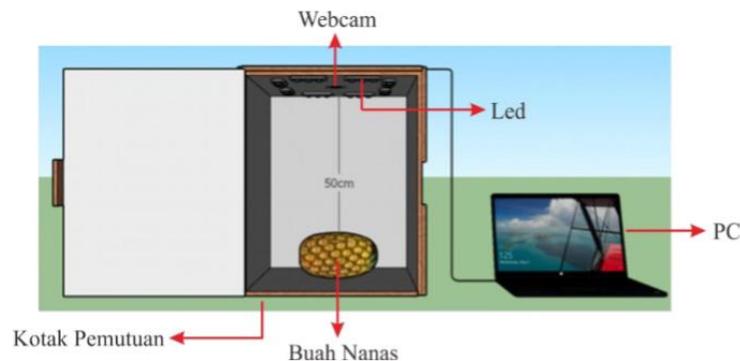
Adapun batasan masalah penelitian ini adalah sebagai berikut.

1. Gambar yang diambil menggunakan webcam Logitech C922 Pro HD Stream dengan resolusi 1080p/30fps dan 720p/60fps.
2. Jenis nanas varietas *Cayenne smooth*.
3. Buah nanas dianggap simetri.
4. Menggunakan metode *thresholding*.

## BAB II TINJAUAN PUSTAKA

### 2.1. Penelitian Terkait

Penelitian mengenai identifikasi kematangan buah nanas secara non-destruktif berbasis *computer vision* (Prasetyo dkk., 2021). Pada penelitian ini dilakukan identifikasi kematangan buah nanas menjadi enam indeks kematangan. Perangkat pendeteksi kematangan buah nanas dapat dilihat seperti Gambar 2.1.

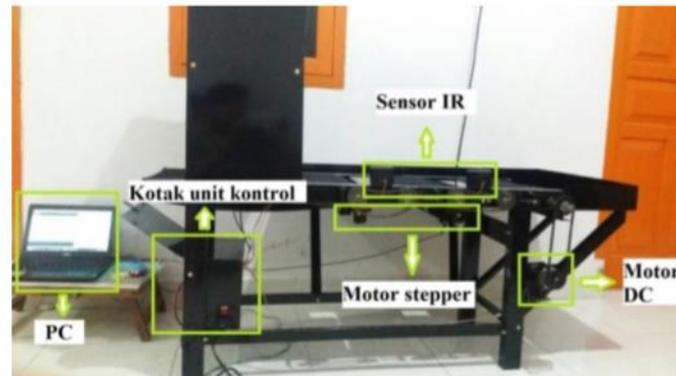


**Gambar 2.1** perangkat pendeteksi kematangan nanas (Prasetyo dkk., 2021).

Pengambilan gambar nanas menggunakan webcam yang ditempatkan di atas kotak, lalu gambar nanas yang didapat akan diolah dengan menggunakan pengolahan citra seperti mengubah gambar nanas ke *grayscale*, lalu gambar dithresholding untuk membuat gambar nanas *grayscale* menjadi biner. Untuk memisahkan objek nanas dari *backgroundnya* dilakukan proses segmentasi.

Proses ini juga berguna untuk mengurangi noise dari gambar. Setelah proses segmentasi kemudian dilakukan ekstraksi ciri. Tujuan dari ekstraksi ciri yaitu untuk memperoleh variabel ciri yang dapat mengklasifikasikan tingkat kematangan nanas. Perancangan jaringan syaraf tiruan dilakukan dalam dua proses, yaitu pelatihan dan analisis kinerja jaringan syaraf tiruan. Masukan dari jaringan syaraf tiruan berupa parameter statistik warna, yaitu standar deviasi, mean, varian, skewness, dan kurtosis dari warna RGB dan HSV buah nanas. Hasil yang diperoleh dari penelitian ini adalah hanya parameter warna dengan korelasi pearson lebih besar dari 0,5 yang digunakan sebagai ciri citra nanas. Keakurasian dari penelitian ini adalah sebesar 98,4% dengan kemampuan mengidentifikasi enam indeks kematangan buah nanas.

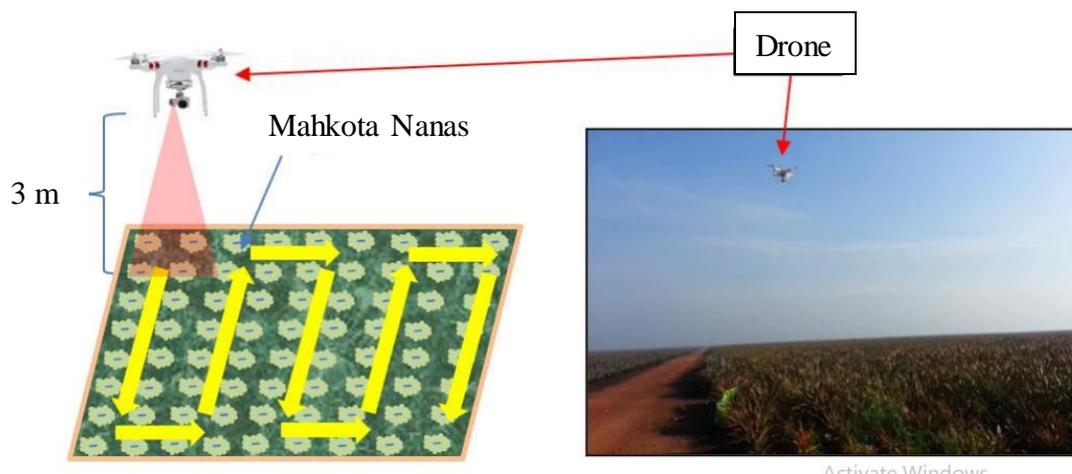
Desy (2020) melakukan rancang bangun purwarupa pemilah nanas berdasarkan tingkat kematangan menggunakan mikrokontroler *Blue Pill STM32F103C8T6* berbasis *computer vision*. Hasil rancangan dari bangun purwarupa ini adalah dapat memilah nanas dengan klasifikasi mentah, matang, dan sangat matang. Metode yang dilakukan pada rancangan pemilahan buah nanas ini terdiri dari sistem mekanik dan rangkaian elektronika untuk menjalankan purwarupa secara otomatis. Pemilahan nanas dilakukan dengan proses simulasi menggunakan data serial indeks kematangan yang dikirimkan dari PC ke mikrokontroler melalui komunikasi serial. Konveyor yang digunakan untuk proses pemilahan menggunakan system transmisi rantai dan roda gigi serta penggerak berupa motor DC. Alat konveyor pemilah buah nanas dapat dilihat pada Gambar 2.2



**Gambar 2.2** Purwarupa pemilah nanas (Desy, 2020).

Dari hasil pengujian yang didapatkan bahwa alat pemilah ini mampu mengangkat buah nanas maksimum sebesar 1 kg dengan kecepatan angkut sebesar 27,98 cm/s dan keakuratan dalam pemilahan kematangan nanas kurang lebih sebesar 90 %.

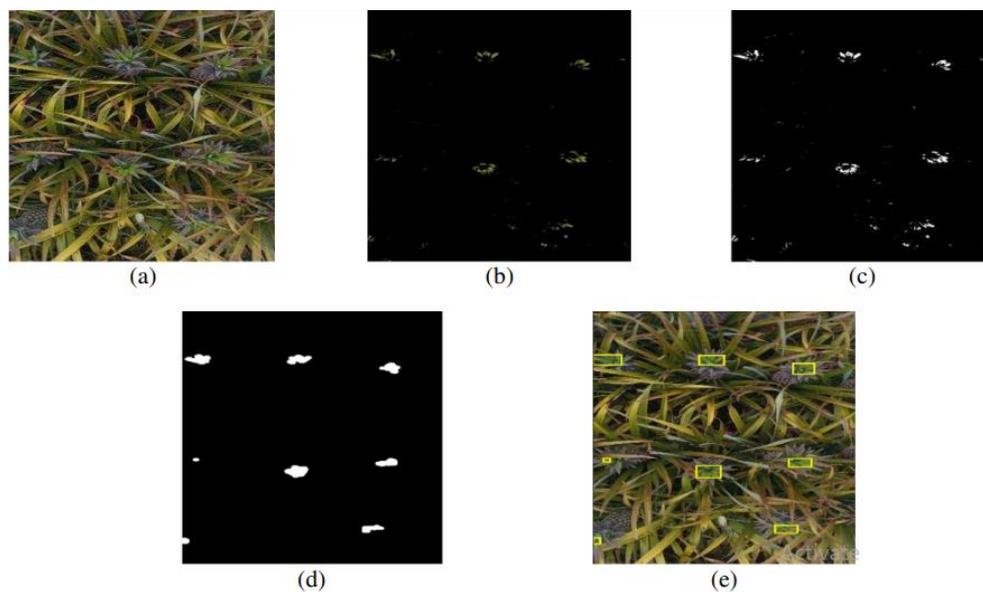
Rahimi (2020) melakukan penelitian dengan menentukan ambang batas (*thresholding*) gambar mahkota nanas dan menghitung jumlah mahkota menggunakan skema transformasi ruang warna. Penelitian ini menggunakan kendaraan udara tanpa awak (UAV) atau biasa dikenal dengan nama drone untuk mengambil citra mahkota nanas di lahan pertanian. Skema deteksi dapat dilihat pada Gambar 2.3



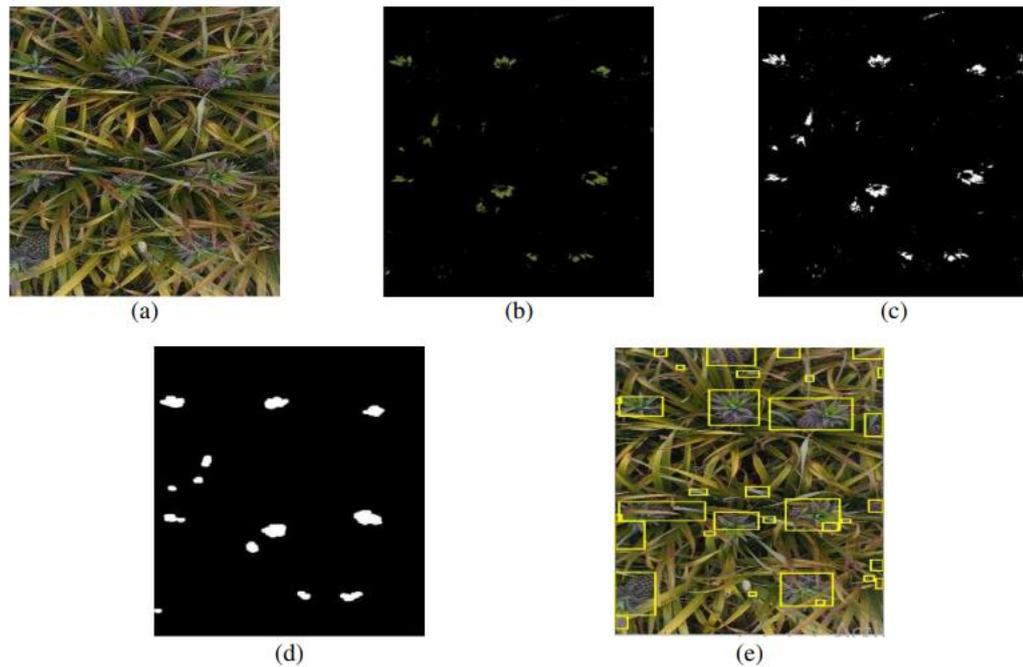
**Gambar 2.3.** Ilustrasi secara real-time dari setup eksperimental di perkebunan nanas (Rahimi dkk., 2020).

Data citra diambil menggunakan drone dengan ketinggian 3 meter. Drone merekam mahkota nanas baris demi baris. Setiap satu baris video akan diekstrak menjadi citra. Fokus utama dari pra-pemrosesan diarahkan untuk mengisolasi bagian mahkota nanas dan mendeteksi jumlah tanaman.

Pada pemrosesan citra menggunakan perbandingan antara dua teknik yang terkait dengan skema ambang batas gambar yang dikenal sebagai skema ruang warna HSV dan  $L^*A^*B^*$ . Kemudian dilanjutkan dengan penentuan ambang batas dengan menerapkan metode *automatic counting* (AC) untuk menghitung hasil panen. Perbandingan hasil deteksi jumlah mahkota nanas dengan menggunakan metode HSV dan  $L^*A^*B^*$  seperti Gambar 2.4 dan 2.5



Gambar 2.4.(a) Gambar hasil ekstraksi; (b) penghapusan latar belakang menggunakan ruang warna HSV; (c) citra biner; (d) operasi morfologi; (e) deteksi mahkota kotak pembatas nanas (Rahimi dkk., 2020).



Gambar 2.5. (a) Gambar hasil ekstraksi; (b) penghapusan latar belakang menggunakan ruang warna  $L * A * B *$ ; (c) citra biner; (d) operasi morfologi; (e) deteksi mahkota kotak pembatas nanas (Rahimi dkk., 2020).

Video yang telah diperoleh selanjutnya diekstraksi ke dalam bentuk gambar, gambar yang diekstraksi mengalami penghapusan latar belakang melalui proses masking yang mempertimbangkan komponen lightness, merah, hijau, kuning dan biru dan thresholding menghasilkan gambar biner yang diinginkan. Selanjutnya, dilakukan operasi morfologi dan prosedur kotak pembatas. Gambar akhir menunjukkan posisi mahkota yang berhasil dideteksi oleh algoritma. Di sini, kinerja teknik penghitungan otomatis yang disarankan dibandingkan dengan menerapkan HSV dan ambang batas warna  $L * A * B *$  untuk penghitungan manual, yang dilakukan oleh pengamat. Peneliti lain telah menerapkan deteksi otomatis dan sistem penghitungan untuk buah-buahan lain seperti apel, jeruk, dll., menghasilkan kinerja yang berbeda yang diukur.

Hasil yang diperoleh adalah ruang warna HSV, menunjukkan pendekatan yang lebih efisien dengan kesalahan rata-rata 12,0% dibandingkan dengan ruang L \* A \* B \* yang memiliki kesalahan rata-rata 59,6%. Dengan catatan bahwa gambar diambil pada ketinggian 3 meter dari tanah dalam kondisi cahaya yang bervariasi nanas (Rahimi dkk., 2020).

## 2.2. Teori Dasar

### 2.2.1 Nanas

Nanas adalah buah yang hidup di daerah tropis berupa tumbuhan semak (Syukur dan Sastrosumarjo, 2019). Asal buah ini yaitu dari negara Brazil, Bolivia, dan Paraguay (Sulistyanto, 2018). Dalam famili nanas terdiri dari 60 genus dengan sekitar 1.500 spesies, klasifikasi tanaman nanas adalah sebagai berikut.

Kingdom	: <i>Plantae</i> (tumbuh-tumbuhan)
Devisi	: <i>Spermatophyta</i> (tumbuhan berbiji)
Sub Devisi	: <i>Angiospermae</i> (berbiji tertutup)
Ordo	: <i>Farinosae (Bromeliales)</i>
Kelas	: <i>Liliopsida (Mokotil berdaun lembaga daun)</i>
Famili	: <i>Bromeliaceae</i>
Genus	: <i>Ananas</i>
Spesies	: <i>Ananas comosus (L) Merr</i>

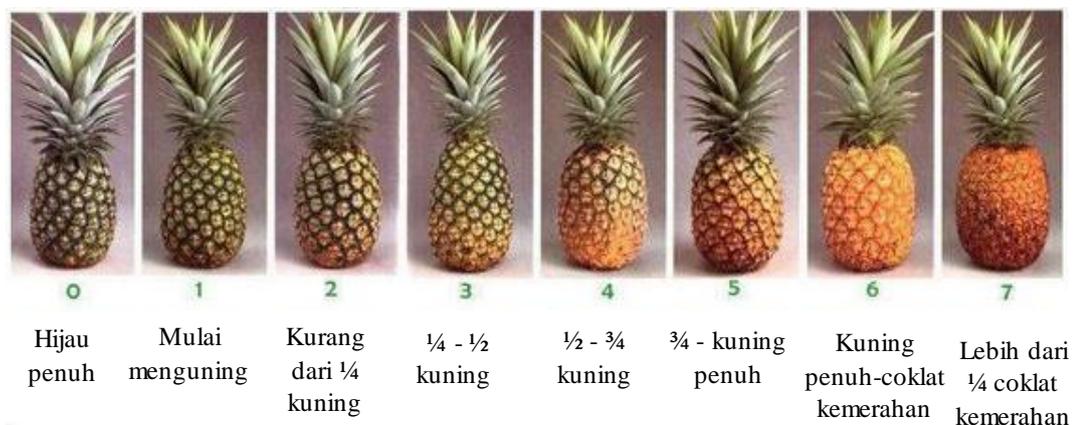
Tanaman nanas dapat tumbuh hingga ketinggian 1-2 m dengan diameter tanamannya kurang lebih 1,5 m. Tanaman ini memiliki batang, dengan batang utama ada di bawah buah yang biasa disebut *butt*. Jumlah daun pada nanas sekitar

68-82 lembar. Susunan daun pada tanaman ini yaitu melingkar, rapi, dan padat berimpitan. Semakin tua usia daunnya, maka daun nanas tersebut berada semakin kedasar tanaman, sedangkan daun muda berada di tengah tanaman. Bentuk dari daun nanas seperti pedang dengan duri di ujung daunnya, tetapi ada sebagian nanas yang tidak memiliki duri pada daunnya. Untuk Tunas utama nanas akan terus tumbuh lurus keatas, bagian tanaman yang tumbuh ini berupa batang dan tunas apeks. Saat tunas apeks telah memproduksi kuncup bunga maka akan tumbuh buah di ujung batang nanas. Bunga nanas tidak memiliki tangkai dan memiliki warna ungu, merah, atau orange. Bunga ini akan tumbuh pada batang utama. Selanjutnya bunga ini akan berkembang menjadi buah nanas (Lubis, 2020). Buah nanas dapat diolah menjadi produk olahan yang sangat enak seperti selai, keripik, sirup, dan lain sebagainya (Harahap dkk., 2019).

Wilayah di Indonesia sangat cocok untuk membudidayakan buah nanas. Misalnya wilayah Lampung, terdapat PT. GGF (Great Giant Foods) yang dulu dikenal dengan nama PT. GGP (Great Giant Pineapple). Produk nanas yang dibudidayakan di perusahaan ini bervariasi *Smooth Cayenne* yang mempunyai karakteristik buah besar dan berat, mata nanas datar atau cawan bunga tidak dalam, daunnya tidak berduri, warna buah pada musim kemarau berwarna kuning dengan kandungan gula yang tinggi, tetapi pada musim hujan buah berwarna kuning pucat (Suryanto, 2019).

Secara umum dalam 100 gram buah nanas *Cayenne smooth* mengandung karbohidrat 11,82 gram, gula 8,29 gram, energi 45 kkal, kalsium 13 miligram, serat 1,40 gram, lemak 0,13 gram, fosfor 9 miligram, protein 0,55 gram, magnesium 12 miligram, potasium 125 miligram, besi 0,25 miligram, seng 0,081 miligram,

thiamine (Vit. B1) 0,078 miligram, riboflavin (Vit. B2) 0,029 miligram, niacin (Vit. B3) 0,106 miligram, pantothenic acid (B5) 0,205 miligram, vitamin B6 0,106 miligram, folate (Vit. B9) 15 miligram, vitamin C 16,9 miligram (Ancos, 2016). Selain melihat kandungan gizi yang cukup tinggi, buah nanas yang memiliki kualitas baik juga membutuhkan tingkat kematangan yang sesuai standar. Tingkat kematangan nanas dapat dilihat pada Gambar 2.6.



**Gambar 2.6.** Tingkat kematangan buah nanas (Mitcham, 2022).

Buah nanas yang berkualitas memiliki tingkat kematangan sesuai dengan permintaan pasar. Pada umumnya nanas mencapai tingkat kematangan antara 120 – 170 hari dihitung dari nanas mulai berbunga. Ada 8 indeks kematangan buah nanas, yaitu dari buah nanas yang berwarna hijau tua atau mentah hingga coklat kemerahan atau ranum. Untuk nanas berindeks 0 diklasifikasikan sebagai nanas muda atau tidak matang dan berwarna hijau tua, nanas tersebut tidak layak untuk dipetik. Buah nanas berindeks 1-2 merupakan nanas yang mulai matang dengan warna hijau tua sedikit kekuningan antara mata dibagian pangkal, nanas ini sudah layak dipetik dan di ekspor. Buah nanas berindeks 3 merupakan nanas yang telah matang dengan keseluruhan mata berwarna hijau dan 25% mata dibagian pangkal

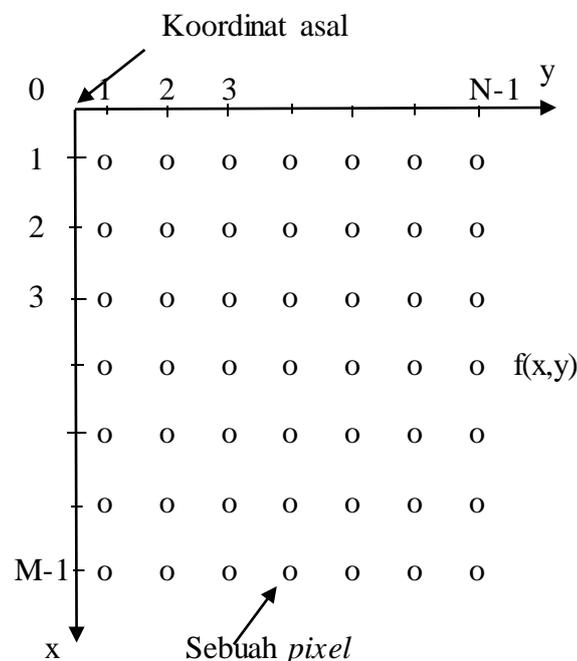
berwarna kuning, untuk nanas indeks 3 juga sudah layak untuk di ekspor. Buah nanas berindeks 4 merupakan buah nanas yang mulai masak dengan sekitar 50%-75% mata di bagian pangkal buah berubah menjadi orange kekuningan. Buah nanas berindeks 5 merupakan buah nanas yang hampir 75%-100% matanya berubah menjadi kuning penuh. Buah nanas berindeks 6 merupakan buah nanas yang telah masak ranum dengan warna kuning penuh-coklat kemerahan. Buah nanas berindeks 7 merupakan buah nanas yang masak ranum mendekati busuk dengan 25 % mata berwarna coklat kemerahan (Mitcham, 2022).

### 2.2.2. Pengolahan Citra

Citra merupakan gambaran atau kemiripan suatu objek. Citra terbagi menjadi dua yaitu citra analog dan citra digital. Citra analog tidak dapat direpresentasikan dalam komputer, sehingga citra ini tidak dapat diproses oleh komputer secara langsung. Untuk dapat diproses maka citra analog harus dikonversi menjadi citra digital. Citra digital sendiri adalah citra yang dapat diolah oleh komputer. Citra yang dihasilkan dari peralatan digital dapat langsung diolah dengan peralatan komputer. Hal ini karena dalam peralatan digital terdapat sistem sampling dan kuantisasi. Sedangkan dalam peralatan analog tidak terdapat kedua sistem tersebut (Andono dkk., 2017). Agar citra dapat diolah dengan komputer, citra harus direpresentasikan secara numerik dengan nilai diskrit. Citra digital ditentukan dengan nilai matrik ukuran  $N \times M$  (Sumijan dan Purnama, 2021).

$$f(x, y) \approx \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0, M-1) \\ f(1,0) & f(1,1) & \dots & f(1, M-1) \\ f(2,0) & f(2,1) & \dots & \dots \\ f(N-1, 0) & f(N-1, 1) & \dots & f(N-1, M-1) \end{bmatrix} \quad (2.1)$$

Untuk sisi kanan dari persamaan (2.1) adalah sebuah citra digital dalam definisi. Dalam sebuah citra terdapat array atau elemen yang biasa disebut elemen citra, elemen gambar, atau piksel. Biasanya digunakan huruf  $N$  dan  $M$  untuk menandakan baris dan kolom. Sebuah matrik  $1 \times M$  dinamakan vektor baris sedangkan matrik  $N \times 1$  dinamakan vektor kolom. Untuk matrik  $1 \times 1$  disebut skalar (Siahaan dan Sianipar, 2020). Citra digital merupakan sebuah array (larik) yang berisikan nilai-nilai real maupun kompleks yang direpresentasikan dengan deretan bit tertentu (Munantri dkk., 2019). Koordinat citra digital dapat dilihat pada Gambar 2.7.



**Gambar 2.7.** Koordinat Citra Digital (Nafi'iyah dan Mujilawati, 2018).

Citra digital merupakan sebuah representasi dari fungsi intensitas cahaya didalam bentuk diskrit pada bidang dua dimensi. Sebuah citra terdiri dari sekumpulan piksel (*picture element*) dengan koordinat  $(x,y)$  dan amplitude  $f(x,y)$ . Koordinat  $(x,y)$  menunjukkan letak atau posisi piksel dalam suatu citra. Sedangkan amplitude  $f(x,y)$  menunjukkan intensitas warna citra. Lalu  $N \times M$  merupakan ukuran matrik dalam sebuah citra (Salamah dan Ekawati, 2021).

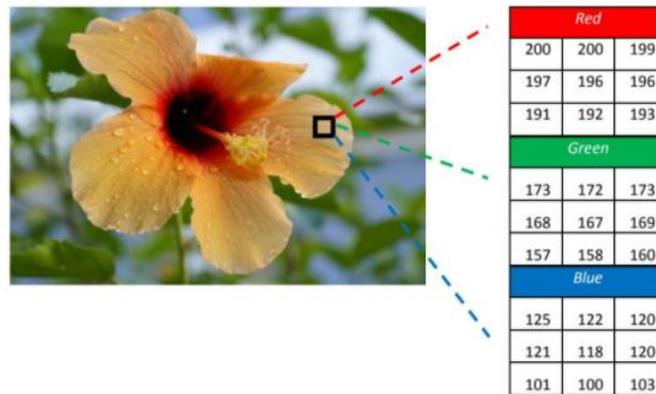
Pengolahan citra merupakan sebuah proses pengolahan piksel-piksel dalam citra digital untuk tujuan tertentu. Ada beberapa alasan dilakukannya pengolahan citra, alasan tersebut antara lain :

1. Agar mendapatkan citra asli dari citra yang banyak noise. Proses pengolahan citra bertujuan untuk mendapatkan hasil citra yang mendekati citra sesungguhnya.
2. Untuk memperoleh karakteristik citra tertentu dan cocok secara visual yang dibutuhkan untuk tahap selanjutnya dalam memproses analisis citra ke proses akuisisi, setelah itu citra akan ditransformasikan ke dalam representasi numerik. Kemudian representasi numerik tersebut akan diproses secara digital oleh komputer.

(Fauzi dkk., 2018).

### **2.2.3. RGB**

RGB (*Red, Green, Blue*) adalah ruang warna yang digunakan untuk menghasilkan citra. Tujuan utama dari warna RGB ini adalah sebagai *sensing, representasi*, dan menampilkan citra dalam sistem elektronik (Lustini dkk., 2019). Citra RGB (*truecolor*) dapat dimodelkan menjadi tiga ruang warna yaitu ruang warna R (*red / merah*), G (*green/hijau*), B (*blue/biru*) sebagai pembentuk komponennya. Format RGB merupakan format standar untuk menampilkan gambar berwarna pada layar komputer (Zonyfar, 2020). Citra RGB ditunjukkan seperti Gambar 2.8



**Gambar 2.8** citra RGB (Fitriyah dan Wihandika, 2021).

Piksel merupakan satuan terkecil data dua dimensi pada citra digital. Citra digital memiliki informasi warna dari intensitas warna yang diterima grid sensor cahaya di dalam kamera. Hasil tangkapan kamera berformat RGB (*Red, Green, Blue*) sehingga setiap piksel memiliki intensitas komponen  $I_R$ ,  $I_G$ ,  $I_B$ . Kemudian masing-masing dari intensitas disusun dalam *regular grid* atau *array 2 dimensi* (Fitriyah dan Wihandika, 2021). Ketiga nilai intensitas tersebut dikombinasikan menjadi sebuah warna baru pada piksel. Misalnya warna kuning sebuah piksel merupakan hasil kombinasi dari intensitas merah 255, intensitas hijau 255, dan intensitas biru 0. Pada contoh lain, untuk mendapatkan warna orange pada sebuah piksel maka dibutuhkan kombinasi dari intensitas merah 255, intensitas hijau 102, dan intensitas biru 0 (Salamah dan Ekawati, 2021).

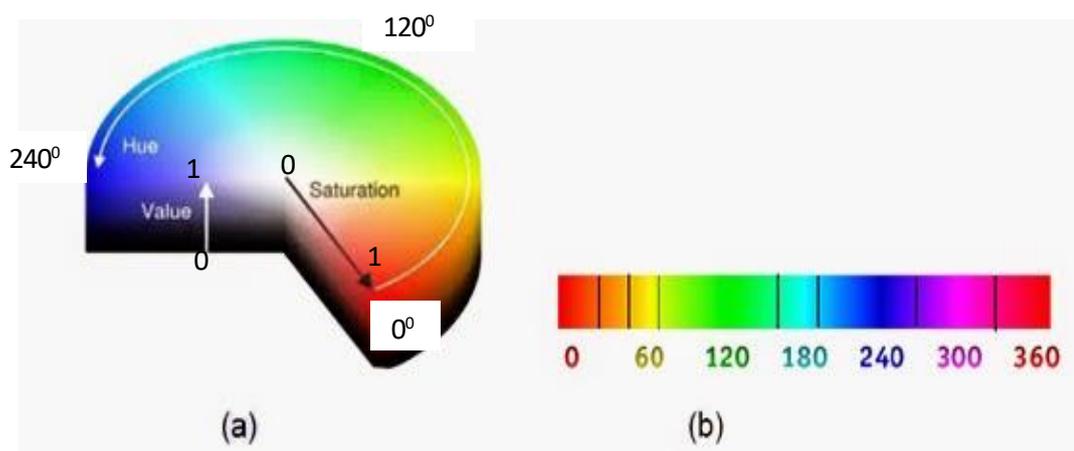
Model warna RGB terdapat tiga komponen yaitu  $r$  sebagai red atau merah,  $g$  sebagai green atau hijau, dan  $b$  sebagai blue atau biru yang merepresentasikan persentase dari sebuah citra digital seperti pada persamaan 2.2.

$$r = \frac{R}{R+G+B}, g = \frac{G}{R+G+B}, b = \frac{B}{R+G+B} \quad (2.2)$$

dengan  $r$  adalah nilai merah,  $g$  adalah nilai hijau,  $b$  adalah nilai biru,  $R$  adalah Intensitas warna merah,  $G$  adalah Intensitas warna hijau, dan  $B$  adalah Intensitas warna biru. Tujuan mencari nilai  $r$ ,  $g$ , dan  $b$  untuk menormalisasi RGB (Pandiangan, 2020).

#### 2.2.4. HSV (*Hue Saturation Value*)

HSV (*Hue Saturation Value*) merupakan ruang warna yang dapat mendeskripsikan warna sesuai mata manusia. Ruang ini memiliki tiga layer yaitu *hue* (derajat warna) menunjukkan derajat warna dengan interval  $[0^\circ, 360^\circ]$  ditunjukkan pada Tabel 2.1, *saturation* (kandungan cahaya putih) direpresentasikan pada sumbu  $y$  dapat dilihat seperti Gambar 2.15, dan *value* (tingkat kemudaan/kegelapan) menunjukkan tingkat kegelapan/kemudaan suatu warna. Semakin muda suatu warna maka *value*-nya semakin tinggi dan semakin gelap suatu warna maka *value*-nya semakin rendah pula (Kirana, 2020). Ruang warna HSV dapat dilihat pada Gambar 2.9



**Gambar 2.9** (a) Ruang Warna HSV, (b) Representasi Hue (Kirana, 2020).

**Tabel 2.1** Representasi Hue

Derajat Warna	Representasi Warna
$0^{\circ}/360^{\circ}$	Merah
$30^{\circ}$	Orange
$60^{\circ}$	Kuning
$120^{\circ}$	Hijau
$180^{\circ}$	Cyan
$240^{\circ}$	Biru
$300^{\circ}$	Ungu

(Kirana, 2020).

Warna kombinasi antara merah, hijau, biru dapat menghasilkan warna kuning, cyan, magenta, dan putih (Kirana, 2020). HSV lebih mendekati mata manusia dibandingkan dengan RGB. Pada ruang HSV, jika nilai R, G, dan B sama maka warna akan berubah menjadi keabuan; yaitu putih. Warna ini hanya mengandung warna putih dan memiliki *saturation* 0. Tetapi bila nilai R, G, dan B berbeda maka *saturation* yang dihasilkan tinggi dan jika nilai R, G, dan B bernilai nol maka nilai *saturation* bernilai 1 (Zonyfar, 2020).

Rumus yang digunakan untuk mencari nilai H, S, V berdasarkan komponen r, g, b yang didapat dari persamaan 2.2. maka dihasilkan rumus seperti persamaan 2.3 dan 2.4.

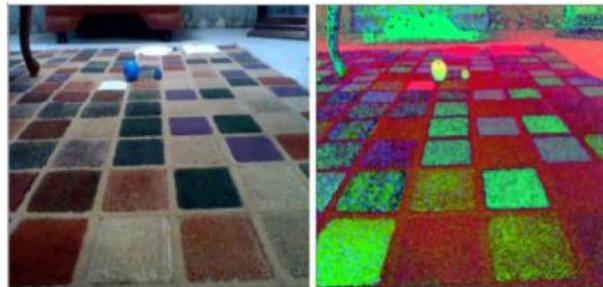
$$H = \begin{cases} 0 & \text{jika } S = 0 \\ \frac{60 (g - b)}{SV}, & \text{jika } V = r \\ 60 \left[ 2 + \frac{b - r}{SV} \right], & \text{jika } V = g \\ 60 \left[ 4 + \frac{r - g}{SV} \right], & \text{jika } V = b \end{cases} \quad (2.3)$$

(2.4)

$$H = H + 360 \text{ jika } H < 0$$

(Kadir, 2019).

Dimana H adalah hue, S adalah saturation, V adalah value, r adalah nilai merah, g adalah nilai hijau, b adalah nilai biru. Contoh hasil konversi citra RGB ke HSV dapat dilihat pada Gambar 2.10.



**Gambar 2.10** RGB ke HSV (Khamdi dkk., 2017).

Citra RGB diubah menjadi HSV sehingga dapat mendeteksi citra dengan menentukan saturasi dan kemurnian dari citra tersebut.

### **2.2.5. Thresholding**

*Thresholding* adalah metode segmentasi citra yang dapat memisahkan objek dengan *background* dalam suatu citra berdasarkan tingkat kecerahannya. Untuk citra yang cenderung gelap akan semakin digelapkan (hitam sempurna dengan intensitas 0). Sedangkan citra yang cenderung terang akan semakin diterangkan (putih sempurna dengan intensitas 1). Sehingga citra keluaran dari metode ini berupa citra biner dengan intensitas piksel sebesar 0 atau 1. Setelah citra disegmentasi maka citra biner yang sudah diperoleh dijadikan sebagai *masking*, dengan melakukan proses *cropping* sehingga didapatkan citra asli tanpa *background* atau *background* yang

dapat diubah-ubah (Setiawan dkk., 2019). Gambar proses *thresholding* dapat dilihat pada Gambar 2.11.

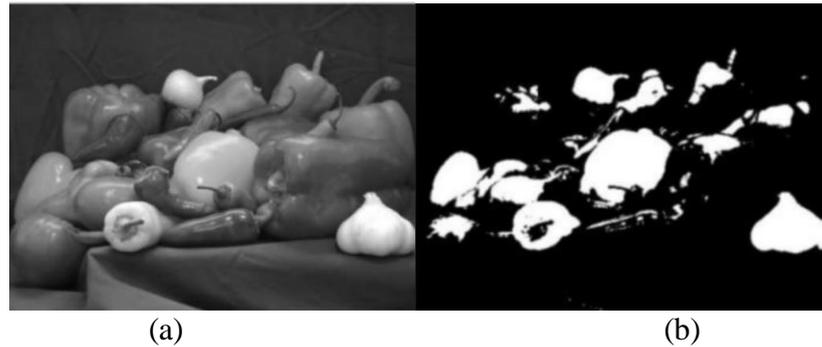
24	21	37	14	8	24	16	29	28	34
9	30	26	10	7	20	23	5	20	31
14	18	4	32	36	13	25	17	26	24
19	11	7	39	43	16	20	26	28	13
6	16	31	13	10	26	13	6	29	11
4	0	23	9	8	24	8	19	22	11
20	0	3	29	34	17	14	38	11	17
31	26	8	34	40	18	26	20	22	45
38	23	26	59	37	21	56	43	55	23
53	2	88	98	72	107	87	82	44	31

↓

0	0	1	0	0	0	0	1	1	1
0	1	0	0	0	0	0	0	0	1
0	0	0	1	1	0	0	0	0	0
0	0	0	1	1	0	0	0	1	0
0	0	1	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	0	0	1	0	0
1	0	0	1	1	0	0	0	0	1
1	0	0	1	1	0	1	1	1	0
1	0	1	1	1	1	1	1	1	1

**Gambar 2.11** Metode *Thresholding* mengubah citra *grayscale* menjadi *biner* (Nafi'iyah dan Mujilahwati, 2018).

Pada Gambar 2.13 menunjukkan proses konversi citra *grayscale* menjadi dalam bentuk biner menggunakan metode *thresholding*. Metode *thresholding* menggunakan nilai ambang  $T$  untuk memisahkan nilai piksel. Nilai *threshold* biasanya dihitung dengan persamaan  $T = (f_{maks} + f_{min}) / 2$ , dimana  $f_{maks}$  sebagai nilai intensitas maksimum dan  $f_{min}$  sebagai intensitas minimum pada citra (Sutoyo, 2009; Heryanto dkk., 2020). Pada Gambar 2.13 menunjukkan nilai ambangnya sebesar 28, bila nilai piksel dibawah nilai ambang akan di anggap sebagai background atau 0 (hitam) dan bila nilai piksel sama atau diatas nilai ambang maka nilai piksel akan dikonversi menjadi objek atau 1 (putih). Pada Gambar 2.12 merupakan hasil *thresholding* citra *grayscale* menjadi citra biner.



**Gambar 2.12.** (a) citra *grayscale*, (b) citra biner (Salamah dan Ekawati, 2021).

Gambar hasil dari thresholding biasanya digunakan lebih lanjut untuk pengenalan objek dan ekstraksi fitur. Metode thresholding secara umum dibagi menjadi dua yaitu:

1. Thresholding Global

*Thresholding global* digunakan dengan mempartisi histogram yang ditentukan dengan nilai *threshold* (ambang batas) *global thresholding* berlaku untuk semua bagian citra.

2. Thresholding Adaptif

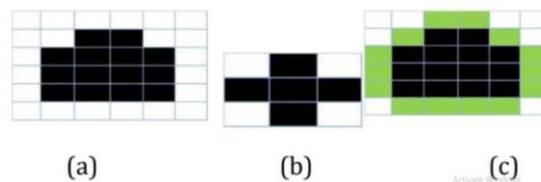
Metode thresholding adaptif dilakukan dengan membagi citra dengan beberapa sub citra. Kemudian, untuk setiap sub citra di lakukan segmentasi dengan nilai *thresholding* yang berbeda (Maria dkk., 2018).

### 2.2.6. Morfologi Citra

Transformasi Morfologi merupakan beberapa operasi sederhana yang digunakan untuk mengoreksi bentuk citra berdasarkan matriks elemen struktur dan kernel sebagai penentu sifat operasinya. Ada dua operator dasar dalam morfologi yaitu erosi (*erotion*) dan dilatasi (*dilatation*), lalu terdapat juga kombinasi dari morfologi dasar tersebut seperti pembukaan (*opening*), penutupan (*closing*), gradien, dan

sebagainya (Hangkawidjaja dkk., 2021). Berikut ini adalah penjelasan dari bagian-bagian morfologi citra yaitu.

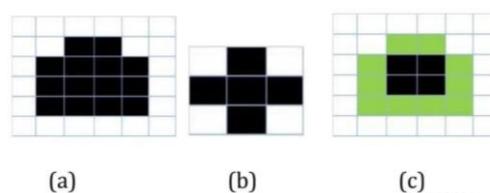
- a. Dilasi atau *dilation* merupakan operasi yang dilakukan untuk memperbesar ukuran segmen dengan menambahkan lapisan disekeliling objek. Ada 2 cara untuk melakukan dilasi yaitu mengubah seluruh titik latar yang bertetangga dengan titik batas menjadi titik objek. Cara kedua yaitu dengan mengubah seluruh titik di sekeliling batas menjadi titik objek. Operasi dilasi dapat dilihat seperti Gambar 2.13.



**Gambar 2.13** (a) citra asli, (b) struktur kernel, (c) citra hasil dilasi (Sumijan dan purnama, 2021).

Pada Gambar 2.13 merupakan contoh dari operasi dilasi. Pada proses dilasi elemen piksel akan dianggap 1 jika setidaknya ada satu piksel yang berisikan dengan kernel 1. Hal ini akan menambahkan wilayah putih pada gambar dan ukuran objek latar depan bertambah.

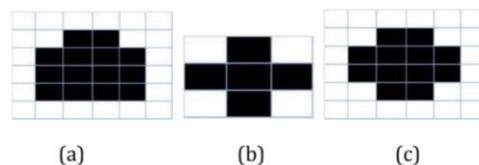
- b. Erosi atau *erotion* adalah kebalikan dari dilasi. Operasi ini akan mengerosi/mengikis permukaan objek dalam citra menjadi lebih kecil. Proses erosi dapat dilihat seperti Gambar 2.14.



**Gambar 2.14** (a) citra asli, (b) struktur kernel, (c) citra hasil erosi (Sumijan dan purnama, 2021).

Pada Gambar 2.14 dapat dilihat bahwa citra asli akan mengalami erosi atau penyempitan. Citra asli tersebut akan dikalikan dengan kernel dan pada proses erosi elemen piksel akan dianggap 1 jika semua piksel yang beririsan dengan kernel bernilai 1. Bila piksel yang beririsan salah satu atau keduanya bernilai 0 maka akan terjadi erosi (penyempitan) (Sutariawan dkk., 2018).

- c. Opening merupakan perpaduan proses citra yang dikenai dengan erosi dan dilanjutkan dengan dilasi. Tujuan dari proses dilasi yaitu untuk memperhalus batas-batas objek, memisahkan objek yang bergandengan, serta menghilangkan ukuran objek yang lebih kecil daripada strukturalnya. Operasi opening ditunjukkan pada seperti Gambar 2.15.

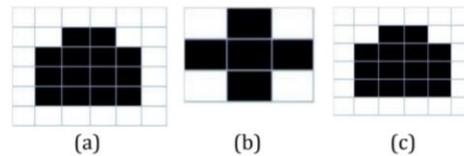


**Gambar 2.15** (a) citra asli, (b) struktur kernel, (c) citra hasil opening (Sumijan dan purnama, 2021).

Pada Gambar 2.15 dapat dilihat bahwa proses opening dari citra asli dengan *structuring element* (struktur kernel) yang telah ditentukan. Operasi ini dilakukan dengan melakukan operasi erosi kemudian hasil dari operasi erosi tersebut di dilasi.

- d. Closing merupakan operasi kombinasi dengan suatu citra dikenai dengan operasi dilasi dan dilanjutkan dengan operasi erosi. Tujuan dari operasi closing yaitu memperhalus objek dengan menyambungkan pecahan-pecahan dan

menghilangkan lubang-lubang kecil pada objek. Operasi closing dapat dilihat seperti Gambar 2.16



**Gambar 2.16** (a) citra asli, (b) struktur kernel, (c) citra hasil closing (Sumijan dan purnama, 2021).

Pada Gambar 2.16 dapat dilihat bahwa proses closing dari citra asli dengan *structuring element* (struktur kernel) yang telah ditentukan. Operasi ini dilakukan dengan mendapatkan hasil dilasi terlebih dahulu dengan melakukan dilasi. Kemudian hasil dari operasi dilasi tersebut di erosi (Sumijan dan purnama, 2021).

### 2.2.7. Operator *Bitwise* dan *Masking*

Operator bitwise digunakan untuk mengoperasikan bit per bit dalam nilai integer. Bitwise terdiri dari operator AND, OR, NOT, OR, XOR, Shl, Shr, Type: int atau char. Biasanya operator ini digunakan untuk memanipulasi data bertipe bit dari data yang bertipe integer atau byte. Tabel 2. Menunjukkan operator *bitwise* dengan simbol-simbolnya.

**Tabel 2.2.** Operator bitwise

Operator	Keterangan
<<	Geser ke kiri
>>	Geser ke kanan
&	AND

	OR
^	XOR
~	NOT

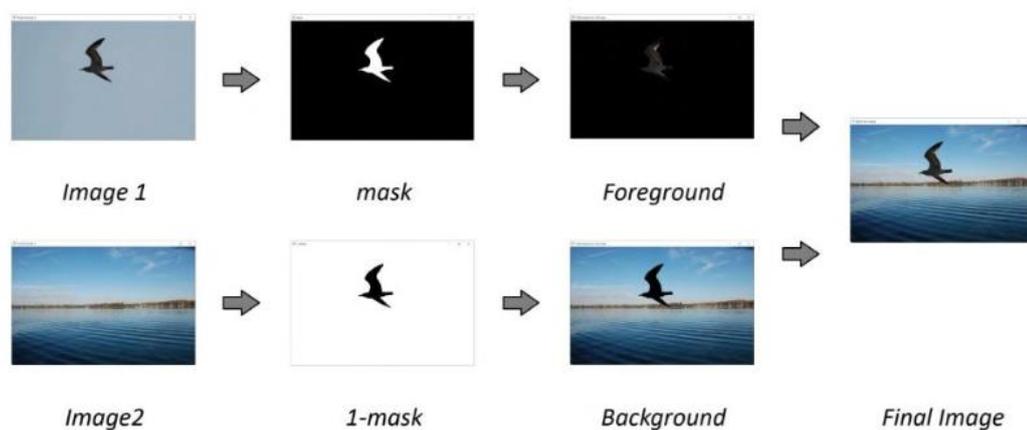
(Novianti, 2020).

*Masking* adalah memadukan dua gambar dengan operasi *bitwise*, perpaduan ini memungkinkan suatu gambar dapat mengubah-ubah *background* sesuai kebutuhan.

Rumus *masking* dilihat pada persamaan 2.6.

$$g(i, j) = \text{mask} \& f_1(i, j) + (1 - \text{mask}) \& f_2(i, j) \quad (2.6)$$

Dimana  $f_1(i, j)$  dan  $f_2(i, j)$  adalah gambar asli yang akan digabungkan dan  $g(i, j)$  adalah hasil gambar; & adalah operasi bitwise AND. Untuk mempermudah pemahaman mengenai *masking* maka perhatikan Gambar 2.17



**Gambar 2.17.** Proses Masking (Chen, 2021).

Pada Gambar 2.17 terdapat dua gambar asli yang akan dipadukan dengan menggunakan operasi *bitwise*. Gambar asli 1 adalah gambar burung sedangkan gambar asli 2 adalah lautan. Langkah pertama yaitu membuat maks dari gambar

asli 1. Cara untuk membuat mask yaitu gambar dikonversi dulu ke dalam ruang warna HSV, lalu menentukan rentang nilai HSV yang terendah dan tertinggi untuk mengambil objek burung. Setelah mendapatkan nilai terendah dan tertinggi, maka gunakan fungsi `cv2.inRange()` untuk mendapatkan mask. Selanjutnya ubah gambar mask dari *grayscale* ke ruang warna BGR. Berikut adalah bagian kode programnya.

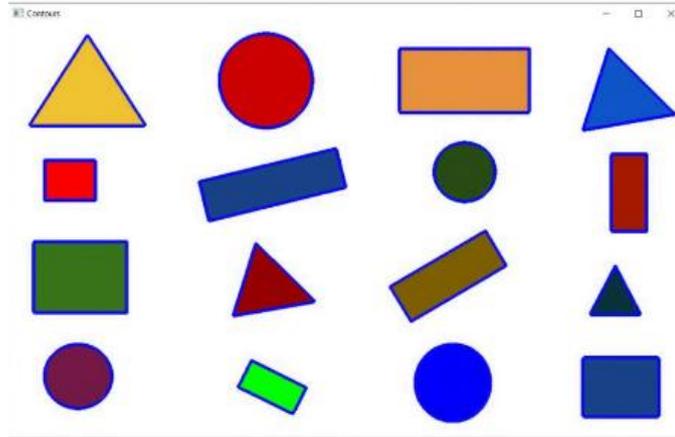
```
imgHSV = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)
mask = cv2.inRange(imgHSV, hsv_lower, hsv_upper)
mask = 1 - mask
mask = cv2.cvtColor(mask, cv2.COLOR_GRAY2BGR)
bg_remove = cv2.bitwise_and(image, mask)
```

Dari program diatas maka didapatkan gambar mask dengan menggunakan operasi *bitwise and*. Latar belakang akan dihapus dan yang tersisa hanya gambar burung saja. Langkah selanjutnya membuat  $(1 - \text{mask})$ , dengan nilai 1 adalah 255. Hasil dari gambar  $(1 - \text{mask})$  dilakukan operasi *bitwise and* dengan gambar 2. Sehingga hanya gambar burung yang dihilangkan sedangkan gambar yang lain dibiarkan. Selanjutnya hasil dari kedua gambar digabungkan menggunakan operasi *bitwise OR*. Maka akan dihasilkan perpaduan gambar 1 dan 2 yang diinginkan (Chen, 2021).

### 2.2.8. Kontur

Kontur dapat dijelaskan secara sederhana sebagai kurva yang menghubungkan semua titik kontinu (sepanjang kontur sebagai alat yang berguna untuk analisis bentuk dan deteksi serta pengenalan objek). Sederhananya, tepi persegi panjang

pada Gambar 2.18 adalah kontur, tepi lingkaran juga merupakan kontur (Chen, 2021).



**Gambar 2.18.** Kontur (Chen, 2021).

OpenCV menyediakan fungsi `cv2.findContours()` untuk mendeteksi kontur dari sebuah gambar. ini mengembalikan array kontur yang terdeteksi dari gambar, setiap kontur disimpan sebagai vektor titik. Fungsi `cv2.drawContours()` digunakan untuk menggambar kontur pada gambar. Sekarang menemukan kontur dari gambar yang telah diproses sebelumnya di bagian atas, lalu menggambar kontur di atas gambar asli, seperti Gambar 2.18. Garis tebal berwarna biru di sekitar bentuk adalah kontur yang telah dideteksi. `cv2.RETR_EXTERNAL` menentukan algoritme pengambilan kontur, khususnya hanya mengambil kontur luar yang ekstrem. `cv2.CHAIN_APPROX_NONE` menentukan algoritme perkiraan kontur, khususnya menyimpan semua titik kontur. Fungsi ini mengembalikan kontur dan hierarki, yang pertama adalah larik yang berisi semua kontur yang terdeteksi dari gambar, yang terakhir berisi informasi topologi gambar, opsional (Chen, 2021).

### 2.2.9. Python dan OpenCV

Program adalah kumpulan perintah manusia yang diberikan kepada komputer untuk melakukan suatu tugas tertentu. Menurut fungsinya perangkat lunak dibagi menjadi tiga yaitu.

- a. Perangkat lunak sistem merupakan perangkat lunak yang bertanggung jawab dalam mengatur komponen perangkat keras (*Hardware*). Contohnya adalah *sistem operasi, kernel, driver*, dan lain sebagainya.
- b. Perangkat lunak programming adalah perangkat lunak yang digunakan untuk menulis program. Contohnya *Netbeans, Pycharm, Anaconda, Eclipse*, dan lain sebagainya.
- c. Perangkat lunak aplikasi adalah perangkat lunak yang digunakan untuk melakukan tugas tertentu. Contohnya *Google Chrome*, dan lain sebagainya.

(Sholihun dan Fatomi, 2021).

Python merupakan sebuah bahasa pemrograman yang interpretative multiguna dengan filosofi dari program ini berfokus pada tingkat keterbacaan kode. Python di klaim sebagai bahasa yang dapat menggabungkan kemampuan, kapabilitas, dengan sintaksi kode yang jelas, dan dengan dilengkapi fungsional pustaka standar yang besar serta komprehensif. Selain itu python juga didukung oleh komunitas yang besar (Rachmatullah, 2020).



**Gambar 2.19** Logo *python* (Rahmadani, 2020).

*Python* memiliki fitur-fitur yang sangat menarik. Salah satu hal yang membedakan bahasa *python* dengan bahasa yang lain adalah sintaksnya (aturan penulisan kode program). Bahasa *python* bisa digunakan hampir di semua sistem operasi. Bahkan digunakan untuk sistem operasi *open source* seperti Linux. Dengan aturan penulisan kode yang simpel dan mudah di implementasikan mengakibatkan banyak para programmer memilih menggunakan bahasa ini. Kelebihan lain dari *python* yaitu merupakan produk *open source* yang dapat diperoleh secara gratis dan dapat digunakan di banyak platform.

Python memiliki beberapa fitur antara lain.

- a. Memiliki library yang luas, *python* menyediakan jenis modul yang bermacam-macam yang siap dipakai untuk berbagai keperluan, bias digunakan secara bebas, dan gratis.
- b. Bahasa ini mudah dipelajari dan memiliki tata bahasa yang jelas dan struktur penulisan kode yang simpel.
- c. Pengecekan, pembacaan kembali, dan penulisan ulang kode sumber yang mudah dipahami serta memiliki aturan kode layout yang jelas.
- d. Dapat mendukung paradigma pemrograman berorientasi obyek.
- e. Mempunyai fitur *garbage collection* yang dapat memudahkan dalam sistem pengelolaan memori otomatis, seperti JAVA.
- f. Modular, bahasa *python* mendukung pemrograman secara modular sehingga mudah dikembangkan dengan menciptakan modul-modul baru, modul-modul tersebut dikembangkan dalam bahasa pemrograman *python* maupun pemrograman C/C++.

g. Sebagai salah satu dari bahasa pemrograman tinggi, python dapat dipelajari dengan mudah karena telah dilengkapi dengan manajemen memori otomatis. Python memiliki fasilitas pengaturan penggunaan *memory* komputer hingga programmer tidak perlu melakukan proses pengaturan *memory* dan bisa langsung menggunakannya (Ilham, 2020; Rahmadhika dan Thantawi, 2021).

Python merupakan bahasa bertipe interpreter. Artinya selama proses eksekusi programnya tidak memerlukan proses kompilasi dengan *compiler*. hal ini sangat berguna untuk menghemat selama pengembangan program (Huda dan Ardi, 2020).



**Gambar 2.20** Interpreter (Rangkuti dkk., 2021).

Interpreter dan *compiler* pada dasarnya sama dalam menjalankan eksekusi kode program (*source code*), kemudian diterjemahkan kebahasa mesin (*object code*). Sehingga komputer dapat melakukan intruksi yang diberikan oleh programmer (Kowi, 2017).

OpenCV (*Open Computer Vision*) merupakan sebuah API (*Application Programming Interface*) Library yang telah familiar pada Pengolahan Citra Computer Vision. Computer Vision merupakan salah satu cabang dari Bidang Ilmu Pengolahan Citra (*Image Processing*) yang memungkinkan komputer dapat melihat seperti manusia. Menggunakan vision tersebut, komputer dapat mengambil keputusan, melakukan aksi, dan mengenali terhadap suatu objek (Rahmadhika dan Thantawi, 2021). Jumlah pengguna openCV mencapai lebih dari 47 ribu orang. OpenCV dibangun menggunakan bahasa pemrograman C. Walaupun begitu masih

memungkinkan bila digunakan bahasa *python* sebagai antarmuka untuk mengakses library *openCV* (Kadir, 2019).



**Gambar 2.21** Logo *OpenCV* (Sugianda dan Thamrin, 2019).

Pada *OpenCV* terdapat *library* sederhana dari pengolahan citra digital mempunyai fitur, antara lain: pelacakan wajah, pengenalan wajah, serta dapat mengolah gambar dan video hingga dimungkinkan dapat mengekstrak informasi didalamnya (Rachmatullah, 2020). Algoritma dari *Computer Vision* dapat menghasilkan dan mengolah data, selain itu *OpenCV* tidak hanya membaca serta menyimpan gambar tetapi juga dapat menambahkan beberapa informasi tambahan mengenai fitur-fiturnya. *OpenCV* memuat format gambar BGR bukan RGB (Spinzhevoy dan Rybnikov, 2018).

Penggunaan dari *library OpenCV* sebagai pengolah citra dapat menggunakan webcam. Untuk mengambil citra diperlukan sebuah *library numpy* yang berfungsi untuk menangani *array* pada citra dan *library cv2* untuk mengakses *OpenCV*, sintak dari program berikut yaitu:

```
Import numpy as np
Import cv2
```

Sintak dari program ini dapat digunakan untuk membuat objek *video capture*. Lalu digunakan sebuah kelas *video capture* dari untuk menangkap video dari file video atau kamera. Inisialisasi dari kelas *video capture* merupakan indeks dari kamera

atau nama file video. Indeks dari perangkat file digunakan untuk menentukan kamera yang digunakan dalam mengambil citra, misalnya indeks kamera pertama adalah 0 seperti kode berikut.

```
cap = cv2.VideoCapture(0)
```

Bagian selanjutnya adalah pengulangan atau *loop* yang berfungsi untuk membaca kerangka gambar dari objek *video capture*. Lalu saat menekan tombol apa saja kode akan berhenti. Untuk menampilkan hasil video digunakan perintah `cv2.imshow()`.

Berikut merupakan sintak dari proses *looping*:

```
while (cap.isOpened( )):
    #capture frame by frame
    ret, frame = cap.read( )
    #display the resulting frame
    cv2.imshow("frame", frame)
    k = cv2.waitKey(0) # untuk menentukan kecepatan gerakan
    if k > 0:
        break
```

bagian akhir untuk menutup jendela (windows) yang terbuka digunakan kode program berikut:

```
cap.release( )
cv2.destroyAllWindows( )
```

(Joseph, 2018).

### 2.2.10. Webcam

Webcam merupakan singkatan dari Web camera, yaitu kamera yang dapat mengambil gambar secara langsung (*real-time*) menggunakan halaman Web, aplikasi IM (seperti Yahoo Messenger), ataupun aplikasi telepon video (seperti

Skype)(Zaki, 1999). Webcam mirip seperti kamera digital, perbedaannya dengan kamera digital adalah webcam dirancang agar dapat memproses foto yang lebih kompleks sehingga dapat dengan mudah di upload ke situs-situs yang ada di internet, misalnya media sosial. Webcam berfungsi untuk memudahkan dalam mengolah pesan dengan cepat seperti chat melalui video atau dengan bertatap muka melalui video yang biasa disebut video call. Fungsi webcam selain untuk berkomunikasi dengan orang lain menggunakan video yaitu dapat digunakan sebagai alat untuk mentransfer media sosial secara langsung (Zulkiffi dkk., 2020). Webcam Logitech C922 Pro seperti Gambar 2.22.



**Gambar 2.22.** Webcam Logitech C922 Pro HD Stream (Logitech, 2022).

Webcam Logitech C922 Pro dapat melakukan streaming dengan resolusi full 1080p dengan 30fps atau HD 720p dengan kecepatan 60fps. Webcam ini juga dilengkapi audio no-drop yang handal, autofocus dan bidang pandang 78°. Mikrofon yang digunakan adalah stereo dengan jangkauan maksimal 1 meter. Webcam ini memiliki tinggi 44 mm, lebar 95 mm, tebal 71 mm, panjang kabel 1,5 m, dan berat 162 g. Jenis lensa yang digunakan yaitu kaca. Webcam ini juga telah dilengkapi dengan tripod (Logitech, 2022).

## BAB III

### METODE PENELITIAN

#### 3.1. Tempat dan Waktu Penelitian

Penelitian ini akan dilaksanakan di Laboratorium Elektronika Dasar dan Instrumentasi, Jurusan Fisika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Lampung bulan April sampai bulan September 2022.

#### 3.2. Alat dan Bahan

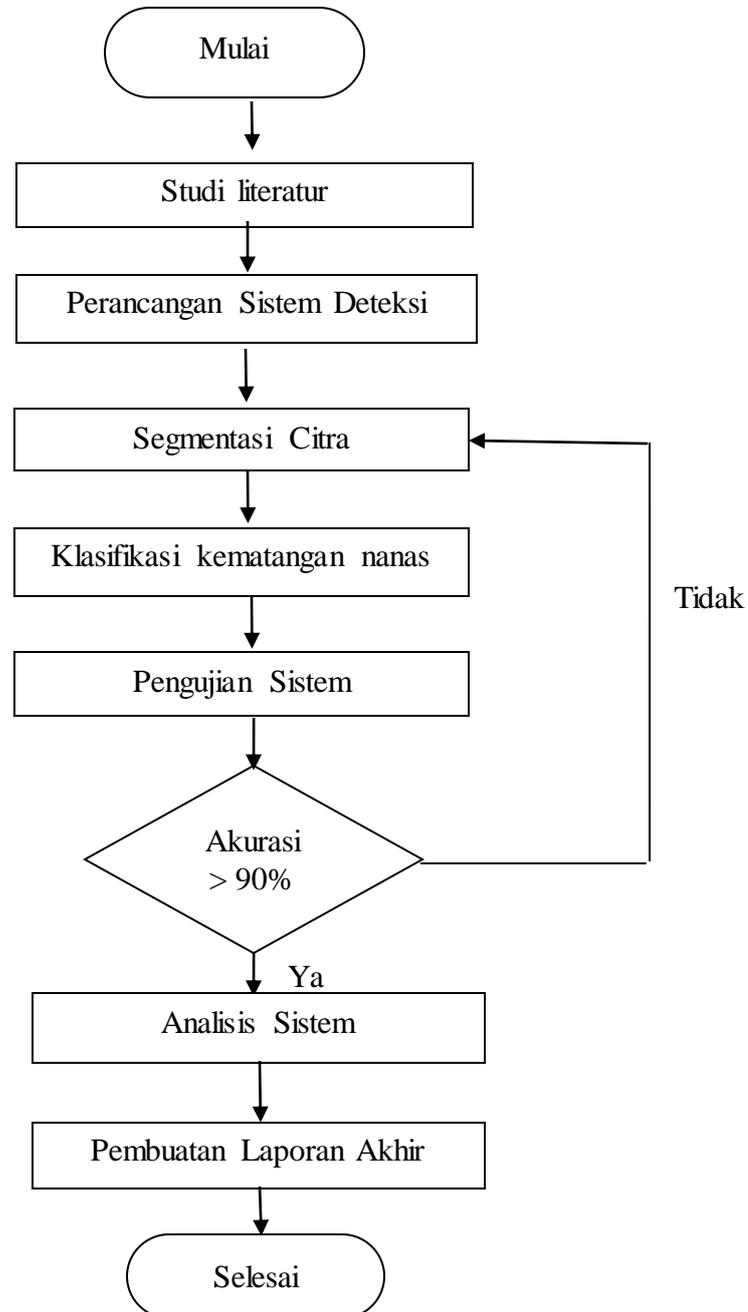
Alat dan bahan yang digunakan pada penelitian ini dapat dilihat pada Tabel 3.1.

**Tabel 3.1.** Alat dan bahan penelitian

No	Nama Alat	Fungsinya
1.	<i>Personal Computer (PC)</i>	Untuk mengolah program mendeteksi kematangan buah nanas.
2.	Perangkat lunak <i>Python versi 3.9</i>	Digunakan sebagai program untuk mendeteksi kematangan buah nanas.
3.	<i>Webcam Logitech C922 Pro</i>	Untuk mengambil gambar sampel nanas.
4.	Kotak pengambilan citra berdimensi 30 x 30 x 50 cm	Untuk tempat mengambil gambar sampel nanas.
5.	Lampu <i>LED strip 117 SMD</i>	Untuk membuat kotak menjadi lebih terang dan gambar yang dihasilkan menjadi lebih baik.
6.	Nana Madu Matang, setengah matang, dan mentah	Untuk sampel uji kematangan buah nanas.

### 3.3. Prosedur Penelitian

Prosedur penelitian secara umum dapat diperlihatkan dalam diagram alir penelitian pada Gambar 3.1



**Gambar 3.1** Diagram Alir Penelitian

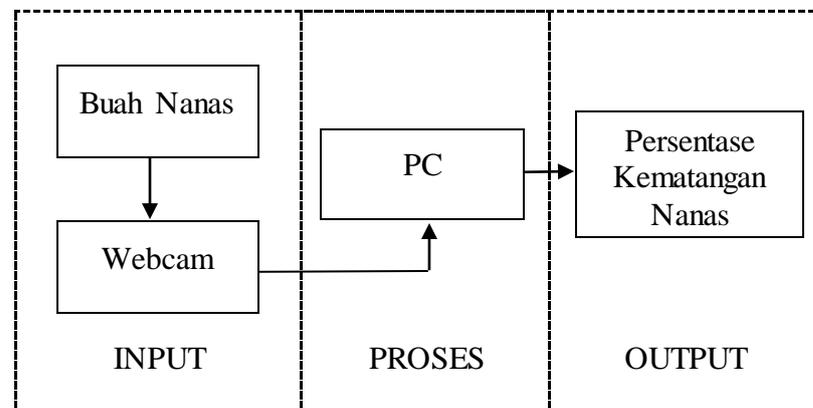
Untuk mencapai tujuan penelitian, prosedur penelitian dijelaskan secara lebih rinci sebagai berikut.

### 3.3.1. Studi Literatur

Studi Literatur dilakukan untuk mengetahui konsep-konsep dasar yang menunjang penelitian ini. Literature yang dikaji pada penelitian ini yaitu pengelompokan tingkat kematangan nanas, teknik pengolahan citra digital, analisis citra, dan konsep dasar dari *thresholding*.

### 3.3.2. Perancangan Sistem Deteksi

Perancangan sistem deteksi buah nanas terdapat dua tahap yaitu perancangan perangkat keras dan perancangan perangkat lunak. Perancangan perangkat keras ditunjukkan pada Gambar 3.2



**Gambar 3.2** Diagram blok sistem identifikasi kematangan buah nanas.

Perancangan perangkat keras tingkat kematangan buah nanas terdiri dari sebuah kotak yang terbuat dari papan kayu berukuran 40 cm x 40 cm x 40 cm sebagai tempat nanas, led sebagai sumber cahaya, webcam untuk mengambil citra buah nanas, dan PC untuk mengolah citra yang telah diambil serta mengidentifikasi

tingkat kematangan buah nanas. Berikut adalah diagram blok perancangan perangkat keras seperti Gambar 3.3.



**Gambar 3.3.** Diagram blok deteksi kematangan buah nanas

Nanas yang digunakan sebagai sampel dimasukkan kedalam kotak pendeteksi kematangan nanas. Setelah itu, webcam akan dihidupkan untuk mengambil citra nanas. Setelah citra nanas didapatkan maka komputer akan mengolah citra nanas menggunakan program *python*. Hasil deteksi nanas akan ditampilkan pada GUI. Hasil deteksi pada GUI dapat menunjukkan jumlah mata nanas yang kuning, jumlah seluruh mata nanas, persentase kematangan, dan klasifikasi tingkat kematangan nanas. Sedangkan desain perangkat lunak pada penelitian ini terdiri dari satu form untuk mengambil gambar, memproses gambar, serta menampilkan hasil deteksi di satu form yang sama seperti Gambar 3.4.

**Gambar 3.4.** Desain *interface* deteksi kematangan buah nanas

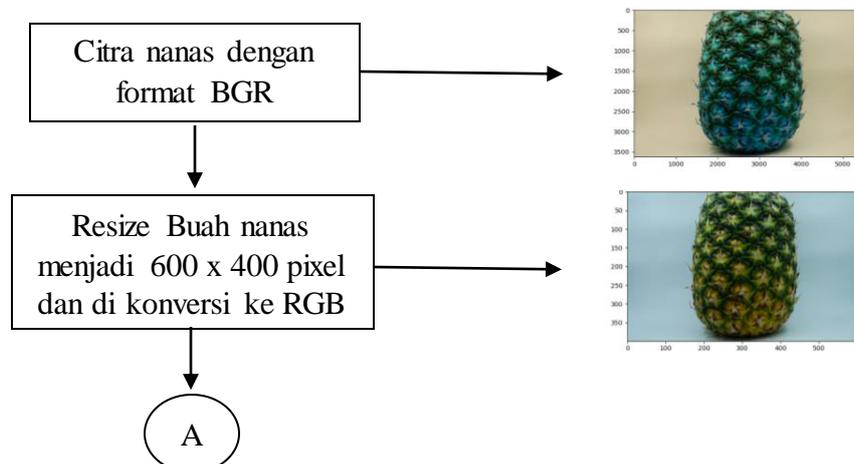
Desain perangkat lunak sistem deteksi kematangan nanas menggunakan aplikasi library PyQt5 dan Qt Designer. Terdapat tiga label utama sebagai penampil video pada webcam, capture gambar, serta menampilkan proses gambarnya. Pada bagian control terdapat 5 push button yaitu tombol start yang berguna sebagai menghidupkan webcam, tombol capture untuk mengambil gambar, tombol save berguna untuk menyimpan gambar yang sudah di capture, tombol proses digunakan untuk mengolah citra nanas, dan tombol exit berguna untuk keluar dari *interface*. Pada bagian hasil menggunakan edit line untuk menampilkan jumlah seluruh mata nanas, jumlah mata nanas yang kuning, persentasi kematangan nanas, dan klasifikasi nanas.

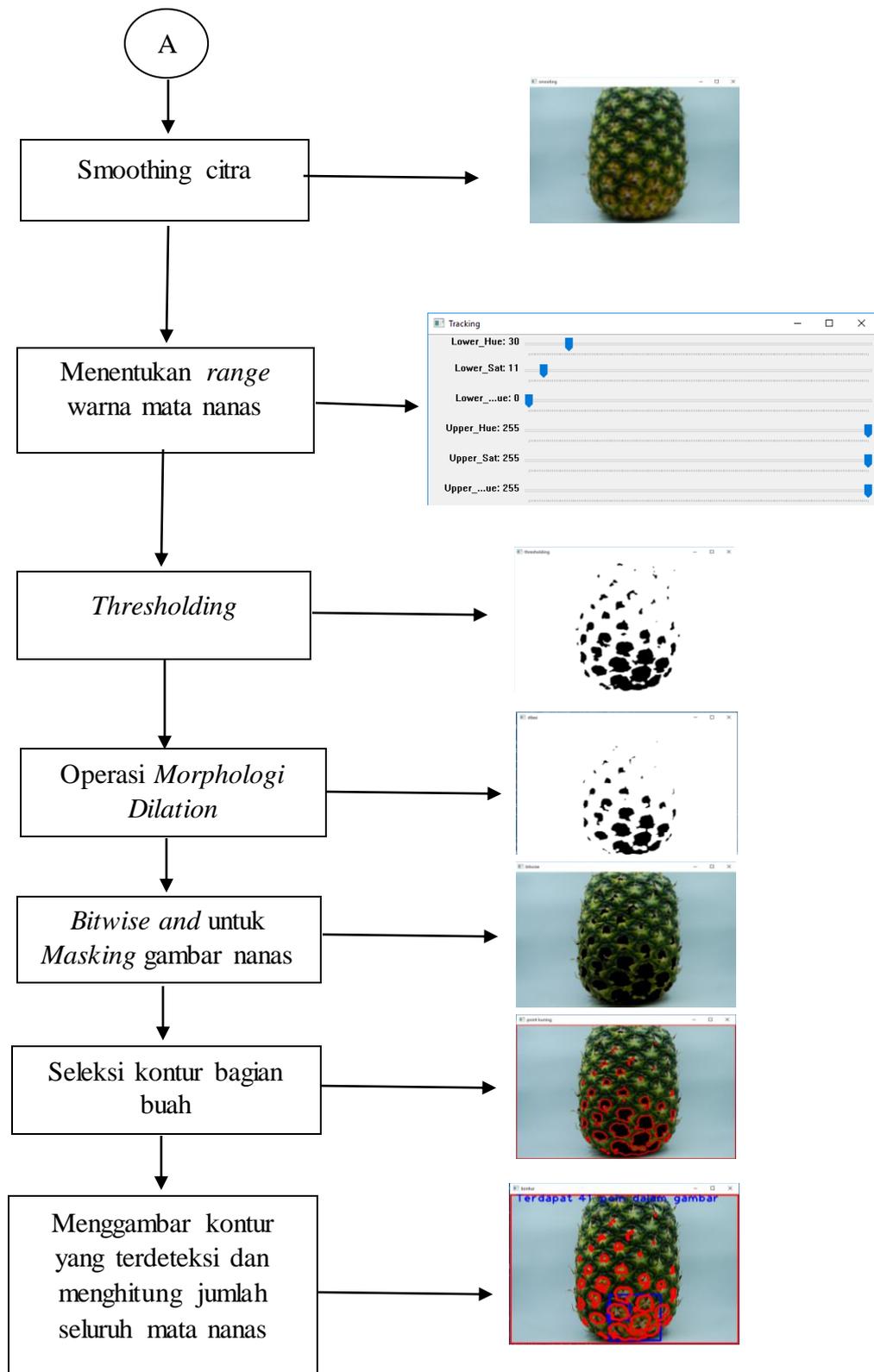
### 3.3.3. Menghitung Jumlah Mata Nanas yang Terdeteksi

Pada penelitian ini dibagi menjadi dua proses deteksi jumlah mata nanas yaitu jumlah seluruh mata nanas dan jumlah mata nanas yang kuning. Proses Segmentasi dan perhitungan jumlah mata nanas adalah sebagai berikut:

#### 3.3.3.1. Menghitung Jumlah Seluruh Mata Nanas

Dalam menghitung jumlah seluruh mata nanas dilakukan dengan menggunakan *computer vision* serta proses segmentasi dilihat pada Gambar 3.5





**Gambar 3.5** Diagram Alir Segmentasi menghitung jumlah seluruh mata nanas

Proses menghitung jumlah seluruh mata nanas dilakukan secara berurutan sebagai berikut:

**Pengambilan citra**, proses ini dilakukan menggunakan webcam yang dikontrol memakai panel GUI, citra yang dihasilkan memiliki format BGR,

**Konversi citra BGR ke RGB**, mengingat citra hasil rekam webcam berformat BGR maka harus diubah menjadi RGB.

**Resize citra**, mengubah citra dari ukuran asli ke ukuran 600 x 400 pixel.

**Smoothing citra**, hasil konversi dan resize citra masih memiliki noise sehingga perlu di *smoothing* agar mengurangi noise tersebut.

**Menentukan ambang batas/threshold**, digunakan taskbar RGB untuk menentukan besar *threshold* yang akan dipakai.

**Threshold**, citra yang telah ditentukan nilai *threshold*-nya di *threshold* untuk memisahkan antara mata nanas dan *background*.

**Dilation**, mata nanas yang telah dipisahkan dari *background* masih ada yang menyatu atau tumpang tindih sehingga perlu dipisahkan dengan *dilation*.

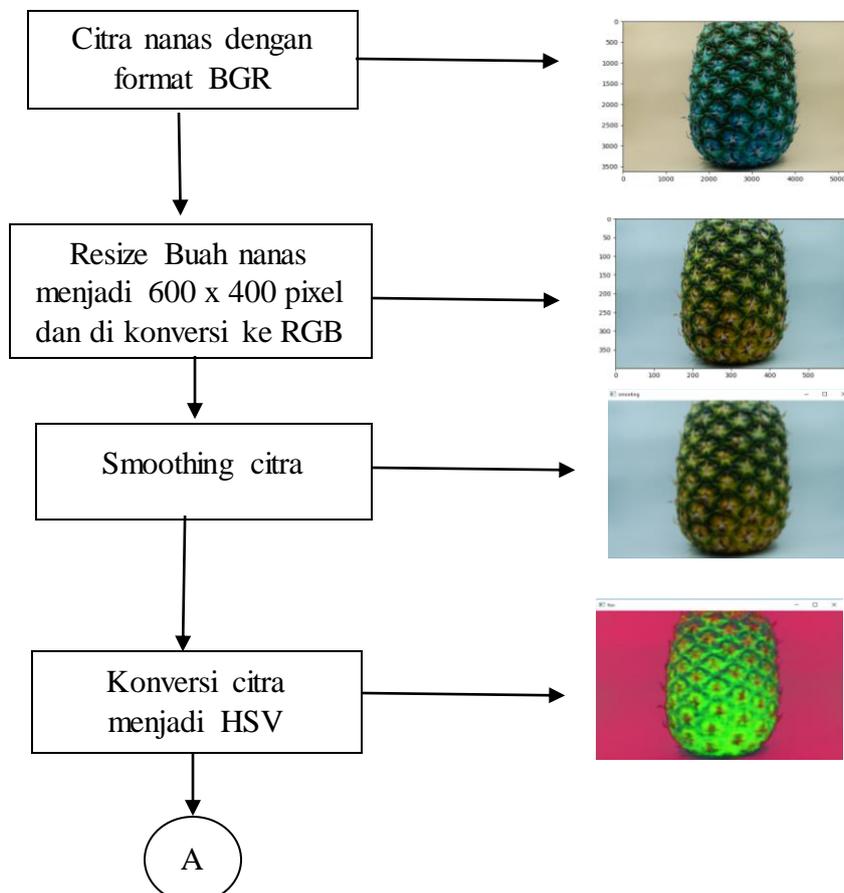
**Bitwise and dan masking citra**, citra hasil *threshold* dikalikan dengan citra asli dengan menggunakan operator *bitwise and* sehingga dihasilkan citra yang telah termasking atau menyatu antara citra hasil *threshold* dengan citra asli.

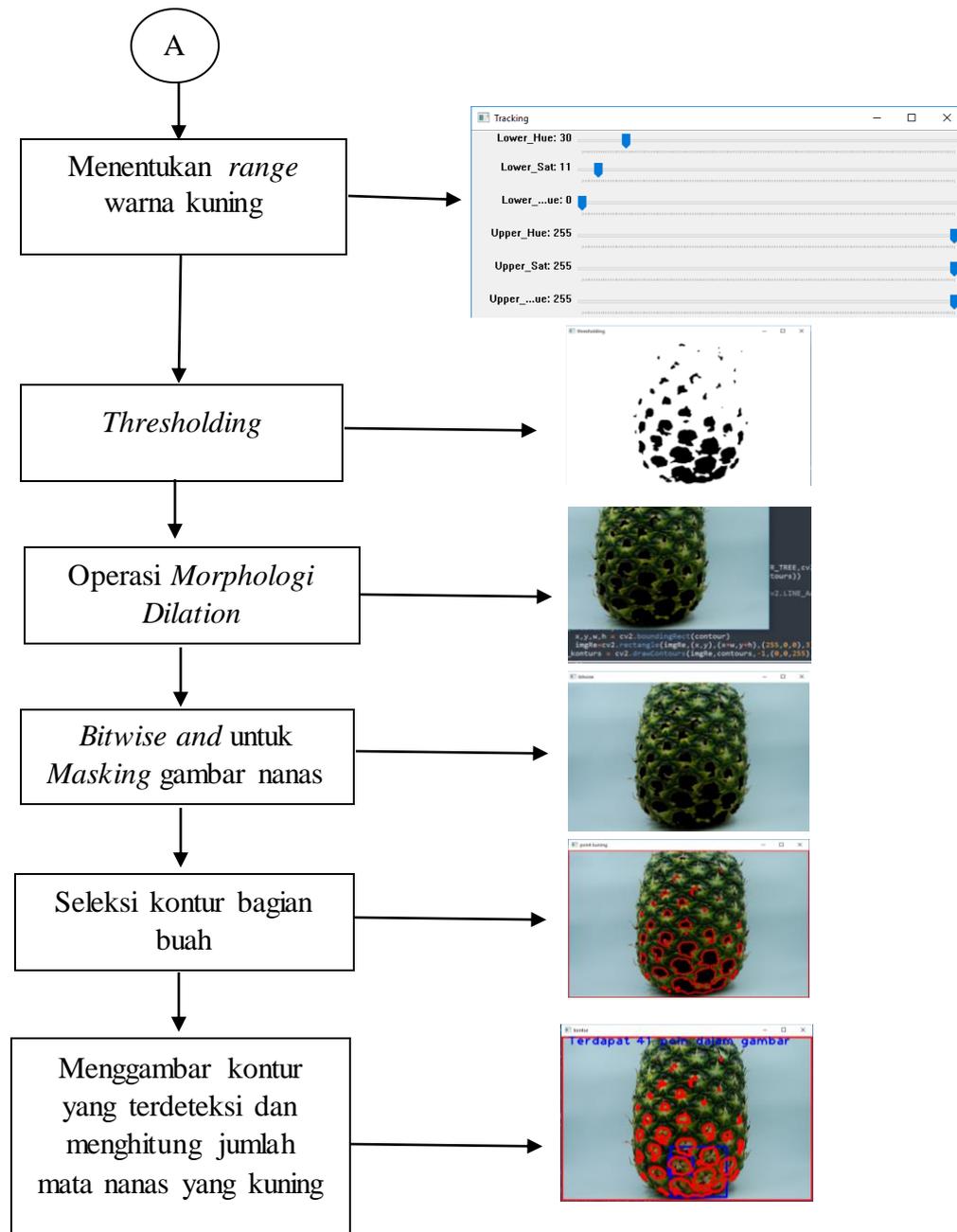
**Seleksi kontur**, setelah citra dilakukan pre-processing selanjutnya dilakukan seleksi kontur, citra akan dikontur pada bagian mata nanas.

**Menggambar kontur dan menghitung jumlah mata,** agar mata nanas yang terdeteksi dapat dilihat maka perlu untuk menggambar kontur. Lalu dilakukan perhitungan jumlah mata nanas dengan menggunakan looping.

### 3.3.3.2. Menghitung Jumlah Mata Nanas yang Kuning

Proses untuk menghitung jumlah mata nanas berwarna kuning langkahnya hampir mirip dengan menghitung jumlah mata seluruhnya, perbedaannya terletak di format yang dipakai sebelum proses *thresholding*, bila format yang dipakai untuk menghitung jumlah seluruh mata nanas berformat RGB sedangkan untuk menghitung jumlah mata nanas kuning menggunakan format HSV. Untuk alur pendeteksi jumlah mata kuning dapat dilihat pada Gambar 3.6





**Gambar 3.6** Diagram Alir Segmentasi menghitung jumlah mata nanas kuning

Proses menghitung mata nanas yang kuning dilakukan secara berurutan sebagai berikut:

**Pengambilan citra**, proses ini dilakukan menggunakan webcam yang dikontrol memakai panel GUI, citra yang dihasilkan memiliki format BGR,

**Konversi citra BGR ke RGB**, mengingat citra hasil rekam webcam berformat BGR maka harus diubah menjadi RGB.

**Resize citra**, mengubah citra dari ukuran asli ke ukuran 600 x 400 pixel.

**Smoothing citra**, hasil konversi dan resize citra masih memiliki noise sehingga perlu di *smoothing* agar mengurangi noise tersebut.

**Konversi citra RGB ke HSV**, citra RGB yang telah di *smoothing* selanjutnya di konversi ke format HSV.

**Menentukan ambang batas/threshold**, digunakan taskbar HSV untuk menentukan besar *threshold* yang akan dipakai.

**Threshold**, citra yang telah ditentukan nilai *threshold*-nya di *threshold* untuk memisahkan antara mata nanas dan *background*.

**Dilation**, mata nanas yang telah dipisahkan dari *background* masih ada yang menyatu atau tumpang tindih sehingga perlu dipisahkan dengan *dilation*.

**Bitwise and dan masking citra**, citra hasil *threshold* dikalikan dengan citra asli dengan menggunakan operator *bitwise and* sehingga dihasilkan citra yang telah termasking atau menyatu antara citra hasil *threshold* dengan citra asli.

**Seleksi kontur**, setelah citra dilakukan pre-processing selanjutnya dilakukan seleksi kontur, citra akan di kontur pada bagian mata nanas.

**Menggambar kontur dan menghitung jumlah mata**, agar mata nanas yang terdeteksi dapat dilihat maka perlu untuk menggambar kontur. Lalu dilakukan perhitungan jumlah mata nanas kuning dengan menggunakan looping.

### 3.3.4. Klasifikasi Kematangan Nanas

Kematangan buah nanas diklasifikasikan berdasarkan persentase tingkat kekuningan dari nanas tersebut. Rumus yang digunakan untuk mengklasifikasikan kematangan mata nanas seperti persamaan 3.1.

$$\% \text{ kematangan nanas} = \frac{\text{jumlah mata kuning}}{\text{jumlah seluruh mata nanas}} \times 100\% \quad (3.1)$$

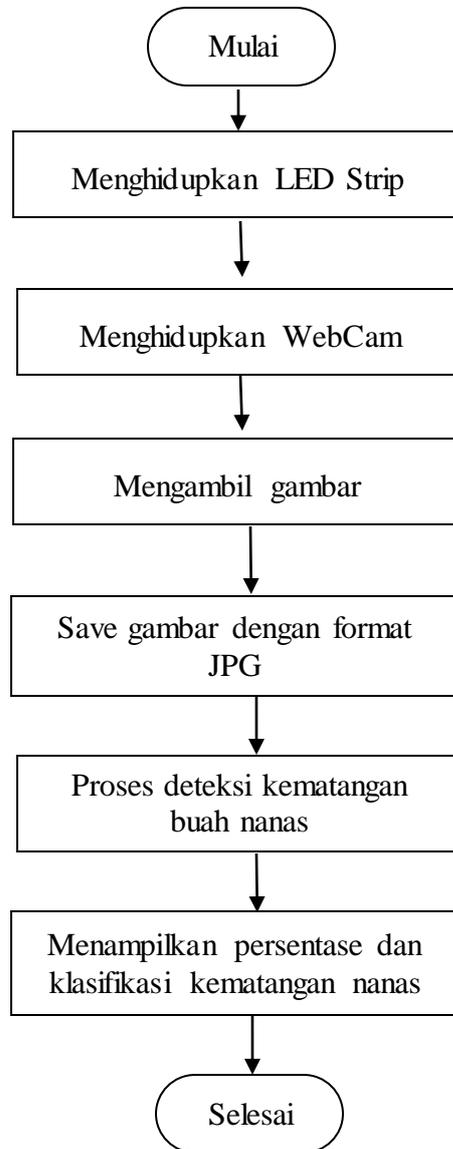
Tingkat kematangan buah nanas dapat dihitung dengan menggunakan rumus jumlah mata nanas yang kuning dibagi dengan jumlah seluruh mata nanas di kali seratus persen. Setelah didapatkan hasil dari persentase kematangan buah nanas maka akan digunakan logika if else untuk mengklasifikasikan kematangan nanas. Untuk membandingkan antara perhitungan manual dengan perhitungan yang dilakukan oleh sistem deteksi maka desain tabel rancangannya dapat dilihat pada Tabel 3.2.

Tabel 3.2. Rancangan tabel data klasifikasi kematangan nanas

Gambar	Manual			Hasil Deteksi			Persentase Kesalahan Kematangan Nanas
	Jumlah Seluruh Mata	Jumlah Mata Kuning	A Persentase Kematangan Nanas	Jumlah Seluruh Mata	Jumlah Mata Kuning	B Persentase Kematangan Nanas	
1							
2							
3							
...							
32							
Rata-rata							

Pada penelitian ini digunakan 32 gambar nanas yang dideteksi dengan metode *thresholding*. Metode ini dapat mendeteksi mata nanas yang selanjutnya menghitung jumlahnya dengan looping. Sistem deteksi kematangan dikontrol dengan panel GUI yang dilihat pada Gambar 3.4. Hasil dari deteksi nanas akan dicatat pada Tabel 3.2. Pada tabel ini pula nilai sesungguhnya dari citra dicatat dengan perhitungan secara manual. Maka akan dihasilkan persentase kesalahan dari

tiap gambar kemudian dirata-rata untuk menemukan nilai rata-rata kesalahan dari sistem. Diagram alir proses pengujian deteksi kematangan nanas dapat dilihat pada gambar 3.7



**Gambar 3.7** Diagram Alir Sistem

Pengambilan data dilakukan dengan menggunakan metode *thresholding* yang dikontrol dan ditampilkan pada panel GUI yang dilihat pada Gambar 3.4., berikut ini adalah proses deteksi kematangan buah nanas dengan metode *thresholding*.

- a. Menyiapkan buah nanas dengan beberapa tingkat kematangan yang berbeda.
- b. Menyiapkan kotak deteksi dengan menghidupkan LED Strip serta menghubungkan webcam ke PC, selanjutnya membuka panel GUI deteksi kematangan nanas.
- c. Setelah alat deteksi dan bahan siap, masukkan nanas ke dalam kotak deteksi.
- d. Untuk menghidupkan webcam maka dapat mengklik tombol “Start”. Akan muncul gambar nanas secara realtime pada layar “Camera”.
- e. Lalu gambar nanas diambil atau ditangkap dengan mengklik tombol “Capture”. Akan muncul hasil tangkapan pada layar “Capture”.
- f. Bila gambar hasil tangkapan ingin disimpan maka dapat mengklik tombol “Save”. Ada beberapa format file yang disediakan untuk menyimpan gambar yaitu jpg, png, jpeg, tiff, dan bmp. Setelah gambar tersimpan maka akan muncul kotak dialog gambar tersimpan.
- g. Gambar nanas akan diproses dengan segmentasi citra yang dapat memisahkan antara mata nanas dengan *background*-nya.
- h. Tunggu beberapa detik sampai proses selesai.
- i. Akan muncul berapa jumlah mata nanas yang kuning, jumlah seluruh mata nanas, dan persentase kematangan nanas. Gambar nanas hasil proses deteksi akan ditampilkan pada layar “Hasil”.
- j. Untuk keluar dari program GUI maka dapat mengklik tombol “Exit”. Maka akan muncul peringatan apakah anda yakin akan keluar atau tidak, bila menekan tombol ya maka pengguna akan keluar dari GUI, sedangkan bila mengklik tidak maka pengguna tidak akan keluar dari tampilan GUI.

Setelah mengetahui jumlah mata nanas kuning, jumlah seluruh mata nanas, dan persentase kematangan nanas maka langkah selanjutnya yaitu menentukan persentase keakuratan dan kesalahan dari gambar yang terdeteksi dengan gambar yang sebenarnya. Rumus untuk melakukan perhitungan seperti persamaan 3.2 dan 3.3.

$$\% \text{ kesalahan} = \frac{|\text{nilai sesungguhnya} - \text{nilai percobaan}|}{|\text{nilai sesungguhnya}|} \times 100 \% \quad (3.2)$$

$$\text{Nilai akurasi} = 100\% - \% \text{ kesalahan} \quad (3.3)$$

(Sulandari dkk., 2014).

### 3.3.5. Pengujian dan Analisis Sistem

Pada tahap ini dilakukan pengujian alat dan analisis sistem deteksi kematangan nanas. Jumlah gambar sampel nanas yang digunakan sebanyak 32 gambar. Setelah diketahui berapa jumlah mata nanas yang kuning, seluruh mata nanas, dan persentase kematangan nanas maka selanjutnya dilakukan perhitungan persentase kesalahan dan akurasi menggunakan Microsoft excel. Rancangan tabel data nilai kesalahan dan akurasi dilihat pada Tabel 3.3.

**Tabel 3.3.** Rancangan tabel data nilai kesalahan dan akurasi sistem deteksi

Gambar	Persentase Kesalahan Jumlah Seluruh Mata	Persentase Kesalahan Mata Kuning	Persentase Akurasi Jumlah Seluruh Mata	Persentase akurasi Mata Kuning	Kesalahan persentase Kematangan Nanas	Akurasi Persentase Kematangan Nanas
1						
2						
3						
...						
32						
<b>Rata-rata</b>						

## V. KESIMPULAN

### 5.1. Kesimpulan

Berdasarkan hasil penelitian dan pembahasan dapat disimpulkan sebagai berikut.

1. Mendeteksi gambar buah nanas berdasarkan persentase mata berwarna kuning dapat digunakan teknik *thresholding*. Pada penelitian ini diperoleh akurasi rata-rata kematangan sebesar  $(93 \pm 2)\%$
2. Instrumentasi untuk mendeteksi kematangan buah nanas dapat dibuat berdasarkan jumlah mata nanas berwarna kuning.

### 5.2. Saran

Saran untuk pengembangan penelitian selanjutnya yaitu

1. Menggunakan jenis kamera beresolusi tinggi, sehingga data yang dihasilkan menjadi lebih akurat.
2. Menggunakan metode CNN untuk menambah keakuratan sistem deteksi kematangan nanas.

## DAFTAR PUSTAKA

- Andono, P. N., Sutojo, T., dan Muljono. 2017. *Pengolahan Citra Digital*. Andi.Yogyakarta.
- Ancos, D. B., Moreno, C. S., dan Aguilar, G. A. G. 2016. *Handbook of Pineapple Technology: Production, Postharvest Science, Processing and Nutrition*. *Handbook of pineapple technology*. Wiley-Blackwell. Chichester.
- Asnor, J. I., Rosnah, S., Wan, Z. W. H., dan Badrul, H. A. B. 2013. Pineapple Maturity Recognition Using RGB Extraction. *International Journal of Electrical and Computer Engineering*, 7 (6), pp. 597-600. doi.org/10.5281/zenodo.1077855
- Azman, Aizuddin, A., dan Ismail, F. S. 2017. Convolutional Neural Network for Optimal Pineapple Harvesting. *Journal of Electrical Engineering*, 16 (2), pp. 1-4.
- Bakar, B. H. A., Ishak, A. J., Shamsuddin, R., dan Hassan, W. Z. W. 2013. Ripeness Level Classification for Pineapple Using RGB and HIS Colour Maps. *Journal of Theoretical and Applied Information Technology*, 57 (3), pp. 587-589.
- Behket, S. dan Ahmed, A. 2017. Graph-based video sequence matching using dominant colour graph profile (DCGP). *Signal, Image and Video Processing (SIViP)*, 12 (2), pp. 291-298. <https://doi.org/10.1007/s11760-017-1157-9>
- Chen, J. 2021. *Learn OpenCV 4.5 with Python 3.7 by Example*. Banes & Noble Press. New York City, AS.
- Desy, F. T., Surtono, A., Apriyanto, A., dan Junaidi. 2020. Rancang Bangun Purwarupa Pemilah Nanas Berdasarkan Tingkat Kematangan Menggunakan Mikrokontroler Blue Pill STM32F103C8T6. *Journal of Energy, Material, and Instrumentation Technology*, 1(3), pp. 81-90. <https://doi.org/10.23960/jemit.v1i3.27>
- Fauzi, J. F., Tolle, H., dan Dewi, R. K.. 2018. Implementasi Metode RGB To HSV pada Aplikasi Pengenalan Mata Uang Kertas Berbasis Android untuk Tuna Netra. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 2 (6), pp. 2319-2325.

- Fitriyah, N., dan Wihandika, R. C. 2021. *Dasar-Dasar Pengolahan Citra Digital*. UB Press. Malang.
- Joseph, L. 2018. *Learning Robotics Using Python Second Edition*. Packt Publishing Ltd. Birmingham.
- Hangkawidjaja, A. D., Pranata, Y. A., dan Jarden, J. J. 2021. *Perancangan Dan Realisasi Detektor Retak Permukaan Dinding Bangunan*. Yayasan Lembaga Gumun Indonesia. Solo.
- Harahap, F., Hasanah, H., Isnani, H., Harahap, N. K., Pinem, M. D., Edi, S., Sipahutar, H., dan Silaban, R. 2019. *Kultur Jaringan Nanas*. Media Sahabat Cendikia. Surabaya.
- Hermawan, A., Effendi, T. R., dan Fadillah, N. 2021. Deteksi Kematangan Buah Pisang Berdasarkan Kulit Menggunakan Metode Multi-Level Thresholding dan YCbCr. *Jurnal Informatika dan Teknologi Komputer*, 2(2), pp. 105 – 108. <https://doi.org/10.33059/j-icom.v2i2.2947>
- Heryanto, I. W. A., Artama, M., Segara, M. W., dan Gunadi, I. G. A. 2020. Segmentasi Warna Dengan Metode Thresholding. *Wahana Matematika dan Sains: Jurnal Matematika, Sains, dan Pembelajarannya*, 14(1), pp. 54-64. <http://dx.doi.org/10.23887/wms.v14i1.23240>
- Hidayat, A. 2020. Perbandingan Metode Canny, Prewitt, dan Sobel Pada Image Jenis-Jenis Alga Untuk Mendeteksi Tepi Citra. *JURIKOM (Jurnal Riset Komputer)*, 7(4), pp. 477-481. <http://dx.doi.org/10.30865/jurikom.v7i4.2263>
- Huda, A., dan Ardi, N. 2020. *Dasar-Dasar Pemrograman Berbasis Python*. UNP Press. Padang.
- Kadir, A. 2019. *Langkah Mudah Pemrograman OpenCV dan Python*. PT. Elex Media Komputindo. Jakarta.
- Kaewapichai, W., Kaewtrakulpong, P., Prateepasen, A., dan Khongkraphan, K. 2007. Fitting A Pineapple Model For Automatic Maturity Grading. *IEEE*, pp. 157-160.
- Khamdi, N., Susantok, M., dan Leopard, P. 2017. Pendeteksian Objek Bola Dengan Metode Color Filtering Hsv Pada Robot Soccer Humanoid. *Jurnal Nasional Teknik Elektro*, 6(2), pp. 123-128. <https://doi.org/10.25077/jnte.v6n2.398.2017>
- Khoiriah, S. 2021. PT GGP Ekspor Buah Segar ke 60 Negara. <https://kupastuntas.co/2021/11/18/pt-ggp-lampung-ekspor-buah-segar-ke-60-negara>. Diakses pada tanggal 4 Maret 2022 pukul 12.19.
- Kirana, K. C. 2020. *Pengolahan Citra Digital Teori Dan Penerapan Citra Digital pada Deteksi Wajah Edisi 1*. Ahli Media. Malang.

- Kowi, E. M. 2017. *Dasar-Dasar Pemrograman Pascal*. Guepedia. Bogor.
- Logitech. 2022. C922 Pro HD Stream Webcam. <http://www.logitech.com/id-id/products/webcams/c922-pro-stream-webcam.html>. Diakses pada tanggal 1 April 2022
- Lubis, E. R. 2020. *Budidaya nanas*. Bhuana ilmu Populer. Jakarta.
- Lustini, A., Saparudin, dan Primanita, A. 2019. Klasifikasi Tingkat Kematangan Buah Nanas Menggunakan Ruang Warna Red, Green, Blue dan Hue-Saturation-Intensity. *Jurnal Digital Teknologi Informasi*, 2(1), pp.1-8. <https://doi.org/10.32502/digital.v2i1.2283>
- Maria, E., Yulianto, Arinda, Y. P., Jumiatty, dan Nobel, P. 2018. Segmentasi Citra Digital Bentuk Daun Pada Tanaman Di Politani Samarinda Menggunakan Metode Thresholding. *Jurnal Rekayasa Teknologi Informasi*, 2(1), pp. 37-47. <http://dx.doi.org/10.30872/jurti.v2i1.1377>
- Mitcham, B. 2022. *Postharvest Handling Banana & Pineapple*. Dept. Plant Sciences UC Davis. San Francisco.
- Munantri, N. Z. 2019. Aplikasi Pengolahan Citra Digital Untuk Identifikasi Umur Pohon. *Telematika*, 16(2), pp. 97-104. <https://doi.org/10.31315/telematika.v16i2.3183>
- Novianti, A. 2020. *Rancang Bangun Sistem Pendeteksi Kebakaran Menggunakan Esp32-Cam Berbasis Iot (Software)*. Politeknik Negeri Sri Wijaya. Palembang.
- Nafi'iyah, N., dan Mujilawati, S. 2018. *Buku Ajar Citra Binarisasi dan Enhancement*. DEEPUBLISH. Yogyakarta.
- Nawawi, M. A. A., dan Ismail, F. S. 2017. Simulation and Segmentation Techniques for Crop Maturity Identification of Pineapple Fruit. *17<sup>th</sup> Asia Simulation Conference*. Faculty of Electical Engineering, Universiti Teknologi Malaysia, Johor, Malaysia.
- Pandiangan, H. S. M. 2020. Segmentasi Citra Untuk Pencarian Kode Warna Cat Menggunakan Metode Thershold HSV. *Bulletin of Information Technology (BIT)*, 1(3), pp. 134-143.
- Pathaveerat, S., Terdwongworakul, A., dan Phaungsombut, A. 2008. Multivariate Data Analysis for Classification of Pineapple Maturity. *Journal of Food Engineering* 89, 112-118. <https://doi.org/10.1016/j.jfoodeng.2008.04.012>
- Prasetyo , N. A., Surtono, A., Junaidi, dan Pauzi, G. A. 2021. Sistem Identifikasi Tingkat Kematangan Buah Nanas Secara Non-Destruktif Berbasis Computer Vision. *Journal of Energy, Material, and Instrumentation Technology*, 2(1), pp. 1-10. <https://doi.org/10.23960/jemit.v2i1.26>

- Purnama, P. A. W., dan Sumijan. 2021. *Teori dan Aplikasi Pengolahan Citra Digital Penerapan Dalam Bidang Citra Medis*. Sumatera Barat: Insan Cendikia Mandiri.
- Rachmatullah, F. M., Hapsari, G. I., dan Meisaroh, L. 2020. Sistem Pengaturan Lampu Lalu Lintas Menggunakan Pengolahan Citra Digital. *e-Proceeding of Applied Science*. 6(2), pp. 3569-3583.
- Rahmadani, S. F. P. E. V. 2020. *BIG DATA: Forecasting Menggunakan Python*. Kreatif Industri Nusantara. Bandung.
- Rahimi, W. N. S., Asraf, H. , dan Ali, M. S. A. M. 2020. Ananas comosus Crown Image Thresholding And Crop Counting Using A Colour Space Transformation Scheme. *TELKOMNIKA Telecommunication, Computing, Electronics and Control*, 18(5), pp. 2472-2479.
- Rahmadhika, M. K., dan Thantawi, A. M. 2021. Rancang Bangun Aplikasi Face Recognition Pada Pendekatan CRM Menggunakan Opencv Dan Algoritma Haarcascade. *Jurnal IKRA-ITH Informatika*. 5(1), pp. 109-118.
- Rahman, M. F. 2016. *Deteksi Dan Pemutuan Nanas Berbasis Computer Vision*. Departemen Teknik Mesin Dan Biosistem Fakultas Teknologi Pertanian Institut Pertanian Bogor. Bogor.
- Rangkuti, Y. M., Idrus, S. I. A., dan Tarigan, D. D. 2021. *Pengantar Program Python*. CV. Media Sains Indonesia. Bandung.
- Salamah, U. G., dan Ekawati, R. 2021. *Pengolahan Citra Digital*. CV. Media Sains Indonesia. Bandung.
- Setiawan, I., Dewanta, W., Nugroho, H. A., dan Supriyono H. 2019. Pengolah Citra Dengan Metode Thresholding Dengan Matlab R2014A. *Jurnal Media Infotama*, 15(2), pp. 65-70. <https://doi.org/10.37676/jmi.v15i2.868>
- Sholihun dan Fatomi, Z. S. 2021. *Pemrograman dan Komputasi Numerik Menggunakan Python*. Gadjah Mada University Press. Yogyakarta.
- Siahaan, V. dan Sianipar, R. H. 2020. *Pemrograman MATLAB Dari Nol Sampai Master Untuk Pemrosesan Citra Digital*. Balige. Sumatra Utara.
- Sumijan dan Purnama, P. W. A. 2021. *Teori dan Aplikasi Pengolahan Citra Digital Penerapan Dalam Bidang Citra Medis*. Insan Cendikia Mandiri. Sumatera Barat.
- Sulandari, W., Hartatik, dan Sudiby, N. A. 2014. *Metode Statistika untuk Kimia: Analisis Data dengan Excel*. BIPTEK INDONESIA. Sukoharjo.

- Sulistiyanto. 2018. *Kiat Hidup Sukses dan Berumur Panjang Meregenerasi Sel-Sel Tubuh Agar Tetap Sehat dan Fit*. Andi. Yogyakarta.
- Sugianda, I. dan Thamrin. 2019. Perancangan sistem Deteksi Objek Pada Robot KRBSI Berbasis Mini PC Raspberry Pi 3. *Jurnal Teknologi Informasi dan Pendidikan*, 12(1), pp. 56-64. <https://doi.org/10.24036/tip.v12i1.177>
- Suryadi, D. 2019. Great Giant Pineapple, Mengekspor Produk ke Lebih dari 60 Negara. <https://swa.co.id/business-champions/great-giant-pineapple-mengekspor-produk-ke-lebih-dari-60-negara>. Diakses pada tanggal 9 Maret 2022 pukul 21.18.
- Suryanto, A. 2019. *Teknologi Produksi Tanaman Budidaya*. UB Press. Malang.
- Susanti, H. 2013. *Pengenalan PT. GGP (Great Giant Pineapple) dan PT. GGLC (Great Giant Livestock) di Terbanggi Besar Lampung Tengah*. Laporan Fieldtrip Bidik Misi. Fakultas Pertanian. Universitas Lampung.
- Syukur, M. dan Sastrosumarjo, S. 2019. *Sitogenetika Tanaman Edisi Kedua*. IPB Press. Bogor.
- Ullah, H., Chaiwong, S., dan Saengrayap, R. 2018. Application of Image Analysis for Maturity Classification of 'Phulae' Pineapple. *The International Conference of Food and Applied Bioscience 2018 proceeding book*, School of Agro-Industry, Mae Fah Luang University, Thailand.
- Zaman, N., Purba, D. W., Marzuki, I., Sa'ida, I. A., Sagala, D., Purba, B., Purba, T., Nuryanti, D. M., Hastuti, D. R. D., dan Mardia. 2020. *Ilmu Usaha Tani*. Yayasan Kita Menulis. Medan.
- Zonyfar, C. 2020. *Pengolahan Citra Digital Sebagai Sebuah Pengantar*. Desanta Muliavisitama. Banten.
- Zulkifli, Hayadi, B. H., dan Ropianto, M. 2020. *Fungsi Dan Manfaat Webcam Pada Windows Pengantar Aplikasi Komputer*. Management Program, University of Ibnu Sina, Indonesia.