

**LAPORAN AKHIR TEKNOLOGI ROBOTIK  
PENGOLAHAN CITRA PRODUK PERTANIAN**



**Muhammad Nibroos Abrar  
F1401211027**

**DEPARTEMEN TEKNIK MESIN DAN BIOSISTEM  
FAKULTAS TEKNOLOGI PERTANIAN  
INSTITUT PERTANIAN BOGOR  
BOGOR  
2024**

## **PRAKATA**

Puji syukur penulis panjatkan ke hadirat Allah SWT atas segala limpahan rahmat dan karunia-Nya sehingga penulisan tugas akhir ini dapat diselesaikan dengan baik. Tugas akhir ini berjudul "Pengolahan Citra Produk Pertanian". Tujuan dari penyusunan tugas akhir ini adalah untuk memenuhi salah satu syarat dalam menyelesaikan mata kuliah Teknologi Robotik.

Ucapan terima kasih yang sebesar-besarnya penulis sampaikan kepada dosen dan asisten praktikum mata kuliah Teknologi Robotik yang telah memberikan bimbingan, arahan, serta dukungan selama proses penyusunan tugas akhir ini berlangsung.

Penulis menyadari bahwa tugas akhir ini masih memiliki kekurangan dan belum sempurna. Oleh karena itu, kritik dan saran yang membangun sangat diharapkan untuk perbaikan dan pengembangan lebih lanjut. Harapan penulis, semoga tugas akhir ini dapat bermanfaat bagi para pembaca dan menjadi referensi yang berguna.

Bogor, Juni 2024

Muhammad Nibroos Abrar

## DAFTAR ISI

<b>PRAKATA.....</b>	<b>i</b>
<b>DAFTAR ISI.....</b>	<b>ii</b>
<b>DAFTAR TABEL .....</b>	<b>iii</b>
<b>DAFTAR GAMBAR .....</b>	<b>iii</b>
<b>BAB 1. PENDAHULUAN .....</b>	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Tujuan.....	2
<b>BAB 2. TINJAUAN PUSTAKA .....</b>	<b>2</b>
2.1 Jeruk .....	2
2.2 Pengolahan Citra Digital .....	2
2.3 Bahasa Pemrograman <i>Python</i> .....	3
2.4 <i>Computer Vision</i> .....	3
2.5 <i>Qt Designer</i> .....	3
<b>BAB 3. METODE.....</b>	<b>4</b>
3.1 Waktu dan Tempat.....	4
3.2 Alat dan Bahan .....	4
3.3 Prosedur .....	4
<b>BAB 4. HASIL DAN PEMBAHASAN .....</b>	<b>5</b>
4.1 Karakteristik Analisis Pengolahan Citra .....	5
4.2 Korelasi Jumlah Piksel dengan Mayor dan Minor Buah Jeruk .....	5
4.3 Korelasi Jumlah Piksel dengan Massa dan Luas Area Buah Jeruk .....	6
4.4 Korelasi Nilai Indeks Warna dengan TPT Buah Jeruk .....	7
4.5 Korelasi Nilai Indeks Warna dengan Kekerasan Buah Jeruk .....	9
<b>BAB 5. PENUTUP.....</b>	<b>10</b>
5.1 Kesimpulan .....	10
5.2 Saran.....	10
<b>DAFTAR PUSTAKA.....</b>	<b>11</b>
<b>LAMPIRAN .....</b>	<b>12</b>
Lampiran 1. Data Hasil Pengukuran Laboratorium Jeruk Mentah .....	12
Lampiran 2. Data Hasil Pengukuran Laboratorium Jeruk Matang .....	16
Lampiran 3. Data Hasil Pengolahan Citra Melalui GUI .....	20
Lampiran 4. Dokumentasi Pengambilan Data Laboratorium .....	23
Lampiran 5. Tampilan Aplikasi Visual Studio Code Dan Qt Designer.....	26
Lampiran 6. Tampilan GUI Pengolahan Citra .....	27
Lampiran 7. Citra Hasil <i>Threshold</i> .....	31
Lampiran 8. Citra Hasil <i>Masking</i> .....	33
Lampiran 9. Kode Pemrograman .....	36

## DAFTAR TABEL

Tabel 1. Data hasil pengukuran laboratorium jeruk mentah .....	12
Tabel 2. Kekerasan jeruk mentah .....	12
Tabel 3. TPT jeruk mentah .....	12
Tabel 4. Hasil perekaman citra jeruk mentah .....	13
Tabel 5. Data hasil pengukuran laboratorium jeruk matang .....	16
Tabel 6. Kekerasan jeruk matang .....	16
Tabel 7. TPT jeruk matang .....	16
Tabel 8. Hasil perekaman citra jeruk matang .....	17
Tabel 9. Hasil pengolahan citra ( <i>RGB, pixel, HSI</i> ) .....	20
Tabel 10. Hasil pengolahan citra (dimensi, brix, kekerasan) .....	21
Tabel 11. Keterangan nama dan fungsi <i>object</i> pada GUI .....	28

## DAFTAR GAMBAR

Gambar 1. Diagram alir prosedur pengambilan data .....	4
Gambar 2. Hubungan jumlah piksel terhadap mayor buah jeruk.....	5
Gambar 3. Hubungan jumlah piksel terhadap minor buah jeruk .....	6
Gambar 4. Hubungan jumlah piksel terhadap massa buah jeruk .....	6
Gambar 5. Hubungan jumlah piksel terhadap luas area buah jeruk.....	7
Gambar 6. Hubungan nilai <i>red</i> terhadap nilai kemanisan buah jeruk .....	7
Gambar 7. Hubungan nilai <i>green</i> terhadap nilai kemanisan buah jeruk .....	8
Gambar 8. Hubungan nilai <i>blue</i> terhadap nilai kemanisan buah jeruk .....	8
Gambar 9. Hubungan nilai <i>red</i> terhadap nilai kekerasan buah jeruk .....	9
Gambar 10. Hubungan nilai <i>green</i> terhadap nilai kekerasan buah jeruk .....	9
Gambar 11. Hubungan nilai <i>blue</i> terhadap nilai kekerasan buah jeruk .....	9
Gambar 12. Pelabelan sampel buah jeruk .....	23
Gambar 13. Pencucian sampel buah jeruk .....	23
Gambar 14. Proses perekaman citra pada sampel .....	24
Gambar 15. Proses pengukuran dimensi sampel buah jeruk .....	24
Gambar 16. Proses pengukuran kekerasan sampel buah jeruk .....	25
Gambar 17. Proses pengukuran massa sampel buah jeruk .....	25
Gambar 18. Aplikasi <i>Visual Studio Code</i> .....	26
Gambar 19. Aplikasi <i>Qt Designer</i> .....	26
Gambar 20. Tampilan awal GUI .....	27
Gambar 21. Tampilan analisis citra GUI.....	27
Gambar 22. Membuka <i>file</i> citra asal .....	29
Gambar 23. Segmentasi citra dengan luaran <i>threshold</i> .....	30
Gambar 24. Segmentasi citra dengan luaran <i>masking</i> .....	30
Gambar 25. Menyimpan <i>file</i> citra hasil.....	31
Gambar 26. Citra hasil <i>threshold</i> mentah 4A.....	32
Gambar 27. Citra hasil <i>threshold</i> mentah 10A.....	32
Gambar 28. Citra hasil <i>threshold</i> matang 1a.....	33

Gambar 29. Citra hasil <i>threshold</i> matang 8b.....	33
Gambar 30. Citra hasil <i>masking</i> mentah 4A .....	34
Gambar 31. Citra hasil <i>masking</i> mentah 10A .....	34
Gambar 32. Citra hasil <i>masking</i> matang 1a .....	35
Gambar 33. Citra hasil <i>masking</i> matang 8b .....	35

## BAB I. PENDAHULUAN

### 1.1. Latar Belakang

Dalam dunia pertanian modern, pemahaman mendalam tentang karakteristik fisik produk pertanian sangat penting untuk meningkatkan kualitas dan efisiensi produksi. Salah satu pendekatan yang telah menjadi semakin populer adalah penggunaan teknologi pengolahan citra digital. Dengan memanfaatkan sistem pengolahan citra yang canggih, para petani dapat dengan mudah menganalisis berbagai aspek fisik produk, seperti ukuran, bentuk, warna, dan kepadatan (Ahmad 2010). Misalnya, teknologi ini dapat digunakan untuk mendeteksi cacat atau kerusakan pada buah-buahan secara otomatis, memungkinkan petani untuk mengidentifikasi produk yang memenuhi standar kualitas yang ditetapkan.

Di sisi lain, pengolahan citra digital juga memainkan peran kunci dalam pemantauan pertumbuhan tanaman dan evaluasi kesehatannya. Dengan memasang kamera pengintai yang terhubung dengan sistem pengolahan citra, petani dapat memantau kondisi tanaman secara *real-time* dan mendeteksi potensi masalah seperti serangan hama atau kekurangan nutrisi. Hal ini memungkinkan mereka untuk mengambil tindakan pencegahan atau intervensi yang diperlukan secara tepat waktu, sehingga meningkatkan produktivitas dan kualitas hasil panen (Satriarini *et.al* 2022). Dengan begitu, kolaborasi antara karakteristik fisik produk pertanian dan teknologi pengolahan citra digital memberikan kontribusi besar dalam menciptakan lingkungan pertanian yang lebih efisien, cerdas, dan berkelanjutan.

Dalam pertanian modern, perbandingan antara data pengukuran langsung dan data pengolahan citra memiliki perbedaan yang signifikan dalam mengoptimalkan produksi dan manajemen sumber daya (Andono dan Sutojo 2018). Data pengukuran langsung, seperti pengukuran fisik manual atau sensorik langsung, memberikan informasi yang penting tentang karakteristik fisik produk pertanian. Namun, pendekatan ini sering kali membutuhkan waktu dan tenaga yang besar, terutama dalam pemantauan yang kontinu dan luas. Di sinilah peran pengolahan citra digital sangat diperlukan. Dengan menggunakan teknologi ini, petani dapat mengumpulkan data secara non-destruktif dan *real-time* tentang produk pertanian, seperti ukuran, warna, tekstur, dan kondisi kesehatan, yang memungkinkan mereka untuk mengambil keputusan yang lebih cepat dan akurat (Hadipernata *et.al* 2020).

Salah satu aplikasi utama pengolahan citra digital pada bidang pertanian pasca panen adalah dalam pemilahan dan sortasi produk. Melalui analisis citra, petani dapat memilah produk-produk dengan cepat berdasarkan karakteristik seperti ukuran, warna, dan keutuhan. Contohnya, sistem pengolahan citra dapat digunakan untuk secara otomatis memilah buah-buahan berdasarkan tingkat kematangan atau deteksi cacat, sehingga meminimalkan kerugian dan meningkatkan efisiensi operasional pasca panen. Selain itu, teknologi pengolahan citra juga digunakan dalam mendeteksi penyakit dan kerusakan pada produk pasca panen. Dengan analisis citra yang canggih, petani dapat mengidentifikasi gejala penyakit, serangan hama, atau kerusakan fisik pada produk secara lebih akurat dan dini. Hal ini memungkinkan pengambilan tindakan pencegahan atau pengelolaan yang tepat waktu, seperti isolasi produk yang terinfeksi atau penggunaan metode penyimpanan yang sesuai, sehingga menjaga kualitas produk secara keseluruhan.

## 1.2. Tujuan

Tugas akhir ini bertujuan melakukan perekaman citra digital terhadap produk pertanian dengan parameter fisik berupa dimensi, massa, kekerasan, dan TPT serta melakukan perekaman citra, melakukan pengolahan citra menggunakan program komputer, dan membandingkan hasil pengolahan citra dengan hasil pengukuran yang sebenarnya, yaitu hasil melalui uji laboratorium.

## BAB II. TINJAUAN PUSTAKA

### 2.1. Jeruk

Jeruk merupakan buah yang populer karena kandungan nutrisinya yang kaya, terutama vitamin C yang berperan penting sebagai antioksidan. Konsumsi jeruk dikaitkan dengan manfaat kesehatan seperti meningkatkan sistem kekebalan tubuh, mendukung kesehatan jantung, dan mengurangi risiko diabetes tipe 2. Selain itu, jeruk juga mengandung vitamin A, vitamin B kompleks, serta mineral seperti kalium dan magnesium yang penting bagi tubuh (Suhaeni 2023). Jeruk memiliki berbagai jenis seperti jeruk Navel, Valencia, Mandarin, dan Lemon, masing-masing dengan karakteristik rasa dan warna kulit yang berbeda. Produksi dan perdagangan jeruk juga menjadi bagian penting dalam ekonomi global, dengan beberapa negara produsen utama seperti Brasil, Amerika Serikat, dan Spanyol. Selain dikonsumsi langsung, jeruk juga sering digunakan dalam masakan dan minuman, menambahkan sentuhan segar dan cita rasa unik pada hidangan. Dengan semua keberagaman manfaat kesehatan, variasi jenis, dan penggunaannya dalam kuliner, jeruk tetap menjadi salah satu buah yang tidak hanya lezat, tetapi juga bermanfaat bagi kesehatan.

### 2.2. Pengolahan Citra Digital

Pengolahan citra digital adalah sebuah disiplin ilmu yang berkembang pesat di bidang teknologi informasi dan komputer. Pada dasarnya, pengolahan citra digital merujuk pada serangkaian teknik komputasi yang digunakan untuk memanipulasi gambar atau citra digital dengan tujuan tertentu (Hadipernata *et.al* 2020). Tahap awal dalam pengolahan citra adalah *preprocessing*, di mana citra disesuaikan, dihilangkan *noise*-nya, dan ditingkatkan kontrasnya agar siap untuk analisis lebih lanjut. Tahap ini sangat penting karena kualitas awal citra dapat memengaruhi hasil akhir dari proses pengolahan. Setelah itu, dilanjutkan dengan tahap *enhancement* yang bertujuan untuk meningkatkan kualitas citra, seperti *sharpening* atau perbaikan kontras.

Selanjutnya, proses pengolahan citra melibatkan segmentasi, di mana objek dalam citra dipisahkan dari latar belakangnya. Hal ini dilakukan dengan teknik-teknik seperti *thresholding*, *clustering*, atau *edge detection*. Setelah objek dipisahkan, dilakukan *feature extraction* untuk menyusun fitur-fitur penting dari citra yang dapat digunakan untuk analisis lebih lanjut atau untuk pengenalan pola (Harani dan Hasanah 2020). Terakhir, tahap *classification* digunakan untuk mengklasifikasikan objek berdasarkan fitur-fitur yang telah diekstraksi sebelumnya.

Meskipun memiliki banyak potensi dan manfaat, pengolahan citra digital juga memiliki tantangan seperti adanya *noise* dalam citra, keterbatasan resolusi,

kompleksitas objek dalam citra yang memerlukan teknik segmentasi yang canggih, dan biaya komputasi yang tinggi karena beberapa proses pengolahan memerlukan sumber daya komputasi yang besar (Mutaqin dan Kom 2023). Dengan pemahaman yang mendalam tentang konsep, aplikasi, metode, dan tantangan ini, pengolahan citra digital dapat terus berkembang dan memberikan kontribusi yang besar dalam berbagai bidang kehidupan manusia.

### **2.3. Bahasa Pemrograman Python**

*Python* adalah bahasa pemrograman yang sangat populer dan mendapat banyak pengakuan di kalangan pengembang perangkat lunak. Sejak awal kemunculannya pada tahun 1991, *Python* telah menjadi salah satu bahasa yang paling diminati karena berbagai keunggulannya. Salah satu keunggulan utama *Python* adalah sintaksisnya yang mudah dibaca dan ditulis, sehingga memudahkan pemula untuk mempelajarinya. Keunikan ini juga membuat *Python* menjadi pilihan yang ideal untuk berbagai keperluan, mulai dari pengembangan web hingga analisis data dan kecerdasan buatan (Rahman *et.al* 2023). Ekosistem *Python* juga sangat kaya dengan berbagai pustaka dan *framework* yang mendukung pengembangan aplikasi yang kompleks dan inovatif.

Meskipun memiliki banyak keunggulan, *Python* juga memiliki beberapa tantangan, terutama terkait dengan kinerjanya dalam aplikasi yang membutuhkan pemrosesan intensif (Hari *et.al* 2023). Meskipun demikian, perkembangan *Python* terus berlanjut dengan rilis versi baru yang memberikan perbaikan performa dan fitur-fitur baru, menjadikannya salah satu bahasa pemrograman yang terus relevan dan diminati di dunia pengembangan perangkat lunak.

### **2.4. Computer Vision**

*Computer Vision* merupakan cabang ilmu dalam bidang kecerdasan buatan yang bertujuan untuk mengizinkan mesin untuk "melihat" dan "memahami" dunia visual seperti manusia. Salah satu aspek penting dari *Computer Vision* adalah deteksi objek, yang merupakan kemampuan sistem untuk mengenali dan mengidentifikasi objek dalam gambar atau video (Arifin *et.al* 2023). Metode yang umum digunakan dalam deteksi objek meliputi *Convolutional Neural Networks* (CNNs), yang telah memberikan kemajuan signifikan dalam akurasi pengenalan objek. Selain itu, segmentasi gambar juga menjadi fokus utama dalam *Computer Vision*, di mana tujuannya adalah untuk memisahkan dan mengidentifikasi setiap objek individu dalam suatu gambar (Rahma *et.al* 2021). Teknik-teknik seperti *semantic segmentation* dan *instance segmentation* digunakan untuk tujuan ini. Dengan kemajuan dalam teknologi komputer dan *deep learning*, *Computer Vision* terus berkembang dan memberikan kontribusi penting dalam berbagai aplikasi, khususnya dalam pertanian.

### **2.5. Qt Designer**

*Qt Designer* adalah alat visual dalam *framework Qt* untuk merancang antarmuka pengguna grafis (GUI) secara intuitif menggunakan *drag-and-drop*. Hal ini memungkinkan pengembang membuat dan mengatur komponen GUI tanpa menulis kode, serta melihat tampilan akhir secara langsung. Alat ini mendukung pembuatan aplikasi lintas platform dengan antarmuka konsisten di berbagai sistem operasi. (Nabijonovich dan Najmiddin 2024).



## BAB III. METODE

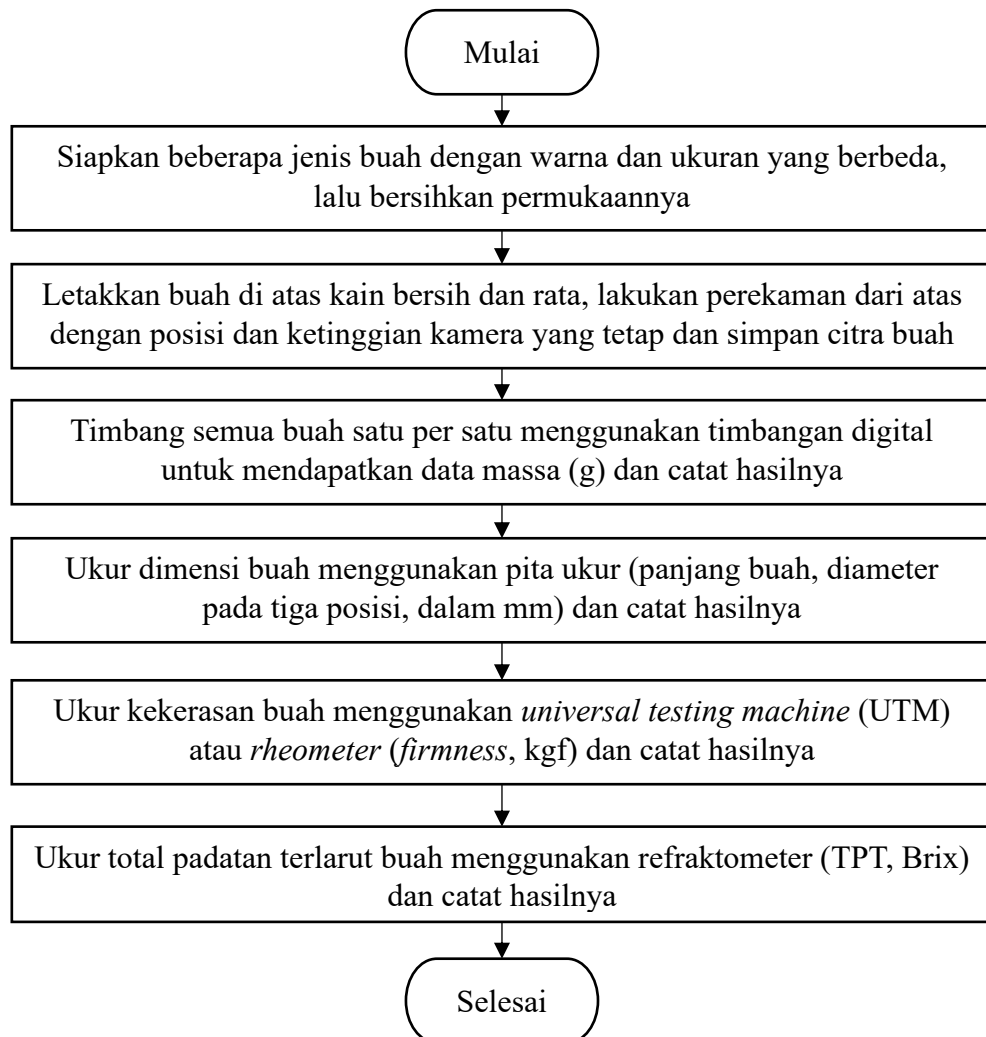
### 3.1. Waktu dan Tempat

Penelitian dilaksanakan selama 6 pertemuan, mulai dari tanggal 19 April 2024 di Laboratorium Instrumentasi dan Kontrol dan Laboratorium Teknik Pengolahan Pangan dan Hasil Pertanian, Departemen Teknik Mesin dan Biosistem, Institut Pertanian Bogor (IPB).

### 3.2. Alat dan Bahan

Alat yang digunakan dalam praktikum ini adalah jangka sorong, timbangan digital, mini studio dan komputer, *rheometer*, *digital refractometer*, pisau, palu, dan pulpen. Adapun bahan yang digunakan adalah 10 buah jeruk matang, 10 buah jeruk mentah, label, akuades, tisu, perangkat lunak Visual Studio Code, Microsoft Excel, *Qt Designer*, dan Bahasa pemrograman *Python*.

### 3.3. Prosedur Praktikum



Gambar 1. Diagram alir prosedur pengambilan data

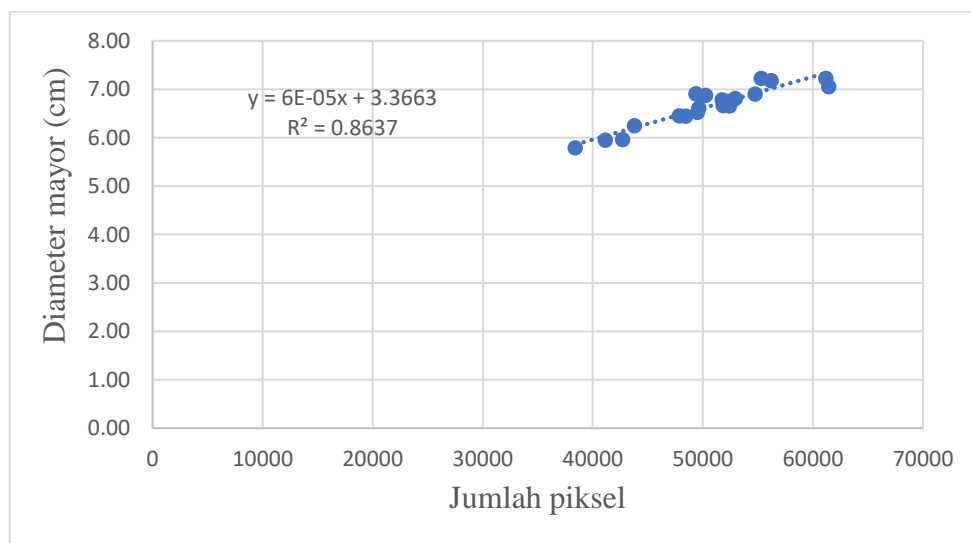
## BAB IV. HASIL DAN PEMBAHASAN

### 4.1 Karakteristik Analisis Pengolahan Citra

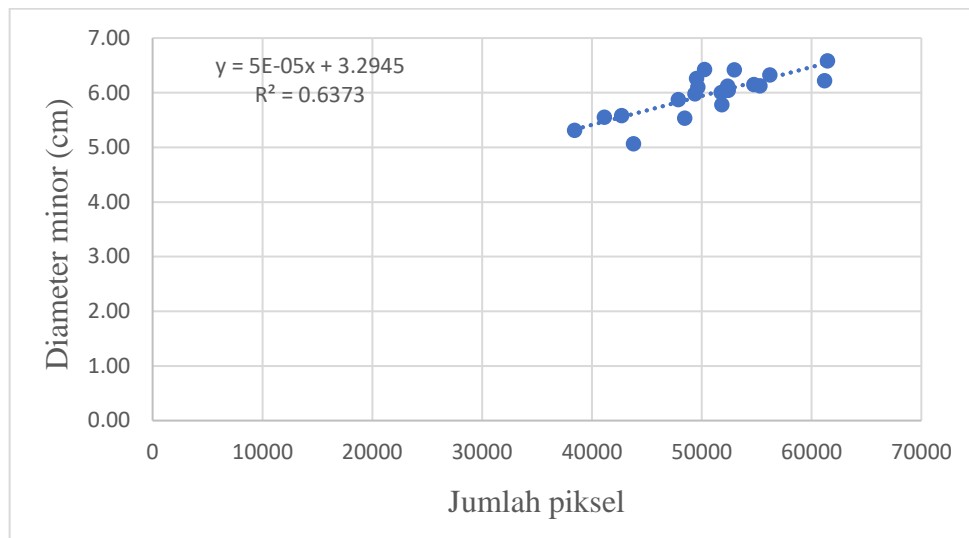
Program pengolahan citra buah jeruk dirancang untuk melakukan pendugaan atau pengolahan citra terhadap citra buah jeruk. Pengolahan citra ini dilakukan dengan menggunakan data primer (pengukuran langsung di lab) 20 sampel buah jeruk dengan dua kelompok berbeda, yaitu 10 sampel buah jeruk mentah dan 10 sampel buah jeruk matang. Pengukuran langsung buah jeruk yang dilakukan antara lain massa dengan menggunakan neraca digital, diameter mayor dan minor dengan menggunakan jangka sorong, kekerasan dengan menggunakan *rheometer*, dan kemanisan menggunakan digital *refractometer*. Selain itu, penelitian ini juga dilakukan pengambilan citra dengan menggunakan seperangkat alat pengolahan citra berupa kamera yang terhubung dengan komputer, di mana setiap buah jeruk dilakukan pengambilan citra sebanyak dua kali dengan posisi buah yang berbeda, yaitu sisi depan dan sisi belakang buah jeruk sehingga diperoleh 40 *file* citra buah jeruk. Kemudian, hasil pengukuran langsung akan digunakan untuk pendugaan pada pengolahan citra.

Program pengolahan citra dirancang dengan menggunakan *software Visual Studio Code* dengan bahasa pemrograman *python* dan tampilan *graphical user interface* dirancang atau didesain dengan menggunakan *Qt Designer*. Program pengolahan citra yang dirancang digunakan untuk menganalisis citra terhadap citra buah jeruk dengan proses *masking*, *thresholding*, pemuaian, dan penyusutan. Variabel atau parameter yang digunakan pada pengolahan citra antara lain indeks warna merah, indeks warna hijau, indeks warna biru, *hue*, saturasi, intensitas atau *value*, dan jumlah piksel. Parameter tersebut digunakan untuk menentukan korelasi antara pengukuran langsung dengan hasil *image processing* dari pengolahan citra. Dengan begitu, selain parameter tersebut, program pengolahan citra yang dirancang juga dapat memberikan keluaran nilai untuk diameter mayor, minor, luas, massa, tingkat kekerasan, dan tingkat kemanisan buah jeruk.

### 4.2 Korelasi Jumlah Piksel dengan Mayor dan Minor Buah Jeruk



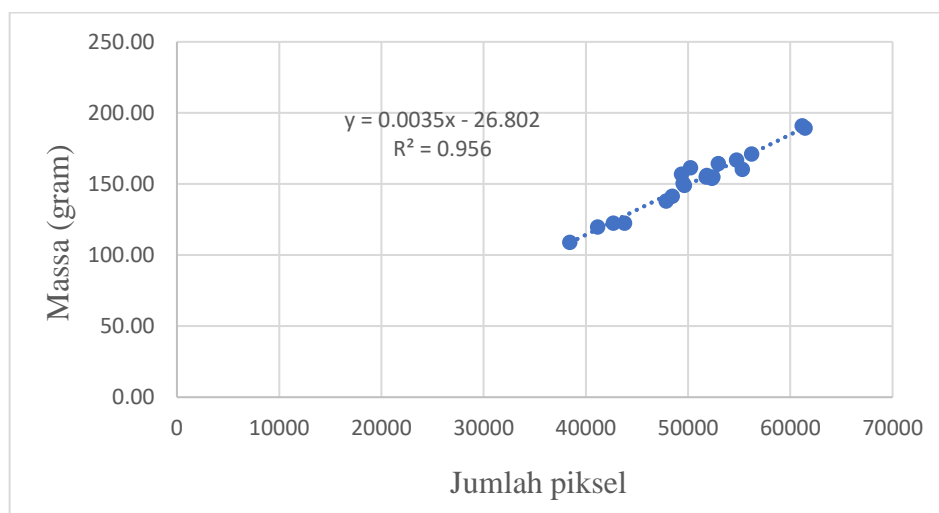
Gambar 2. Hubungan jumlah piksel terhadap mayor buah jeruk



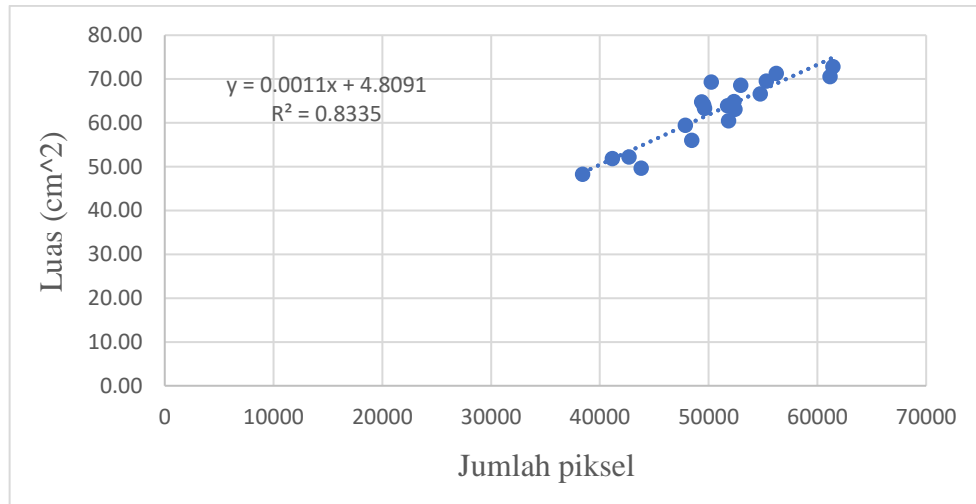
Gambar 3. Hubungan jumlah piksel terhadap minor buah jeruk

Pada program, untuk menghitung diameter mayor dan minor buah jeruk digunakan perhitungan yang melibatkan jumlah piksel gambar buah jeruk. Mula-mula dicari terlebih dahulu korelasi atau hubungan masing-masing antara jumlah piksel dengan diameter mayor dan minor buah jeruk yang telah diukur di lab dengan menggunakan persamaan linear. Gambar 2 dan Gambar 3 menunjukkan bahwa jumlah piksel memiliki hubungan berbanding lurus terhadap diameter mayor dan minor buah jeruk. Hubungan jumlah piksel dengan diameter mayor dan minor menghasilkan nilai *R-squared* masing-masing sebesar 0,8637 dan 0,6373. Nilai *R-squared* menunjukkan tingkat ketepatan hasil prediksi *image processing* berdasarkan jumlah piksel dalam mencari nilai diameter mayor dan minor buah jeruk. Nilai *R-squared* berada pada rentang 0-1, semakin mendekati 1, menunjukkan bahwa prediksi yang dihasilkan semakin mendekati hasil yang sebenarnya sesuai dengan data yang diperoleh dari laboratorium.

#### 4.3 Korelasi Jumlah Piksel dengan Massa dan Luas Area Buah Jeruk



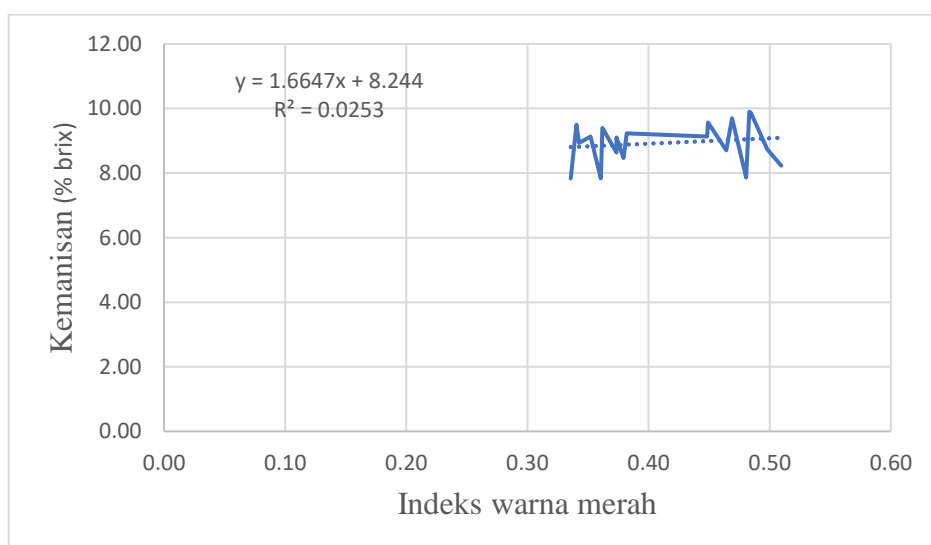
Gambar 4. Hubungan jumlah piksel terhadap massa buah jeruk



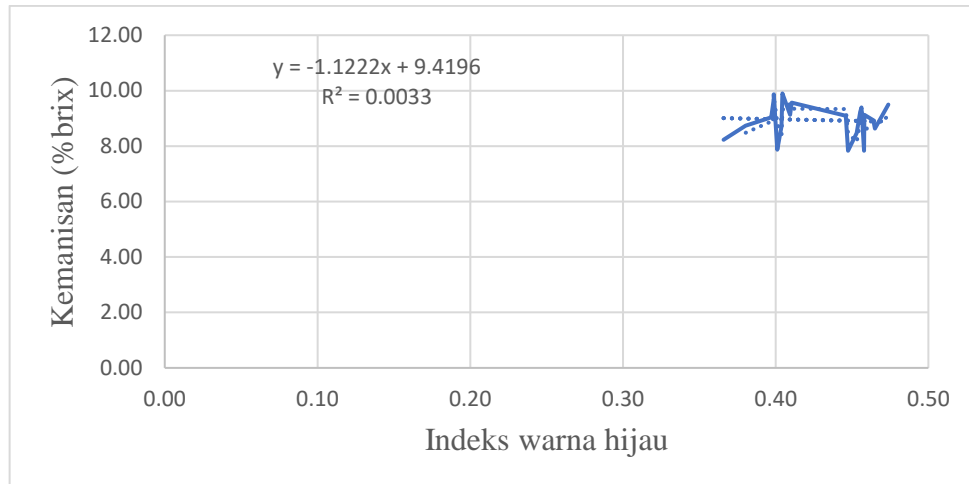
Gambar 5. Hubungan jumlah piksel terhadap luas area buah jeruk

Pada program, untuk menghitung massa dan luas area buah jeruk digunakan perhitungan yang melibatkan jumlah piksel gambar buah jeruk. Mula-mula dicari terlebih dahulu korelasi atau hubungan masing-masing antara jumlah piksel dengan massa dan luas area buah jeruk yang telah diukur di lab dengan menggunakan grafik. Dari grafik pada Gambar 4 dan Gambar 5 tersebut, dapat dilihat bahwa jumlah piksel memiliki hubungan berbanding lurus terhadap massa dan luas area buah jeruk dengan persamaan yang terdapat pada grafik. Pada hubungannya dengan massa buah jeruk, persamaan linear memiliki *R-squared* sebesar 0,956 atau dengan kata lain akurasi yang diperoleh mencapai 95,60%, sedangkan pada hubungannya dengan luas area buah jeruk, persamaan linear memiliki *R-squared* sebesar 0,8335 atau dengan kata lain akurasi yang diperoleh mencapai 83.35%. Besarnya akurasi yang melebihi 80% menunjukkan bahwa persamaan tersebut dapat digunakan untuk menduga massa dan luas area buah jeruk melalui citra buah jeruk.

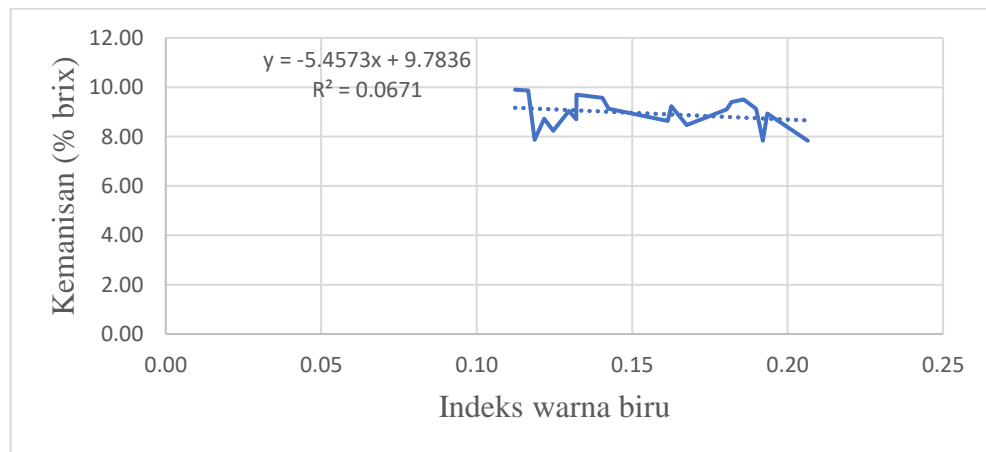
#### 4.4 Korelasi Nilai Indeks Warna dengan TPT Buah Jeruk



Gambar 6. Hubungan nilai *red* terhadap nilai kemanisan buah jeruk



Gambar 7. Hubungan nilai *green* terhadap nilai kemanisan buah jeruk

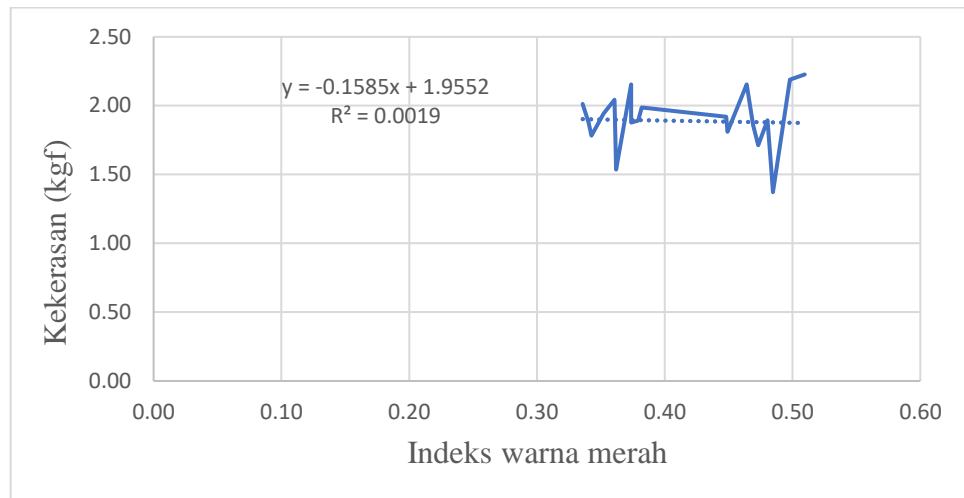


Gambar 8. Hubungan nilai *blue* terhadap nilai kemanisan buah jeruk

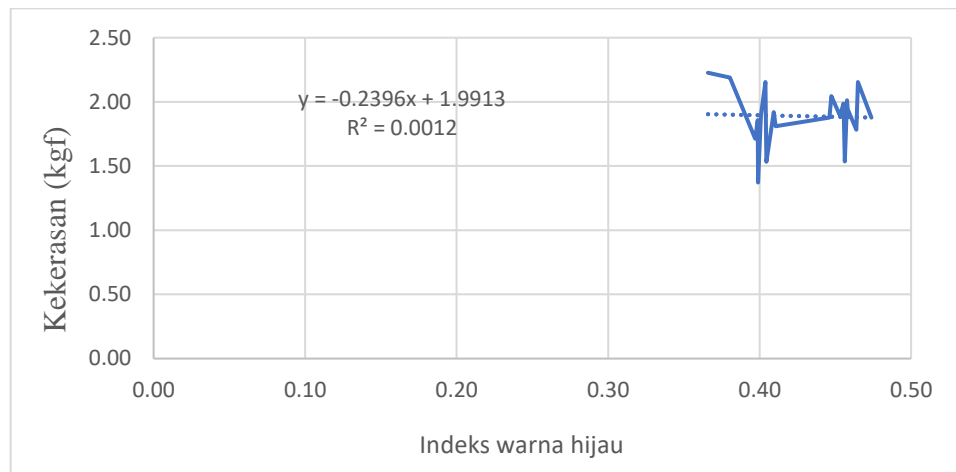
Nilai indeks warna RGB dari setiap citra buah jeruk dianalisis untuk menemukan korelasinya dengan tingkat kemanisan buah jeruk, yang datanya telah diukur sebelumnya di laboratorium dan divisualisasikan menggunakan grafik. Grafik tersebut menghasilkan persamaan linear untuk nilai indeks warna merah, hijau, dan biru, serta nilai akurasi masing-masing yang dinyatakan dalam *R-squared*. Berdasarkan Gambar 6, 7, dan 8, nilai *R-squared* yang diperoleh berturut-turut adalah 0,0253 untuk indeks warna merah, 0,0033 untuk indeks warna hijau, dan 0,0671 untuk indeks warna biru.

Dari nilai-nilai tersebut, indeks warna biru memiliki tingkat akurasi tertinggi dalam memprediksi tingkat kemanisan buah jeruk dibandingkan dengan indeks warna lainnya. Namun, nilai *R-squared* untuk ketiga indeks warna tersebut cukup rendah dan belum cukup memadai untuk digunakan dalam estimasi yang akurat mengenai tingkat kemanisan buah jeruk. Oleh karena itu, prediksi tingkat kemanisan buah jeruk tidak dapat dilakukan secara andal menggunakan parameter nilai indeks warna RGB atau dengan ketiga persamaan yang terdapat pada Gambar 6, 7, dan 8. Perlu dilakukan pengembangan metode atau pencarian parameter tambahan yang lebih efektif untuk meningkatkan akurasi dalam menduga tingkat kemanisan buah jeruk.

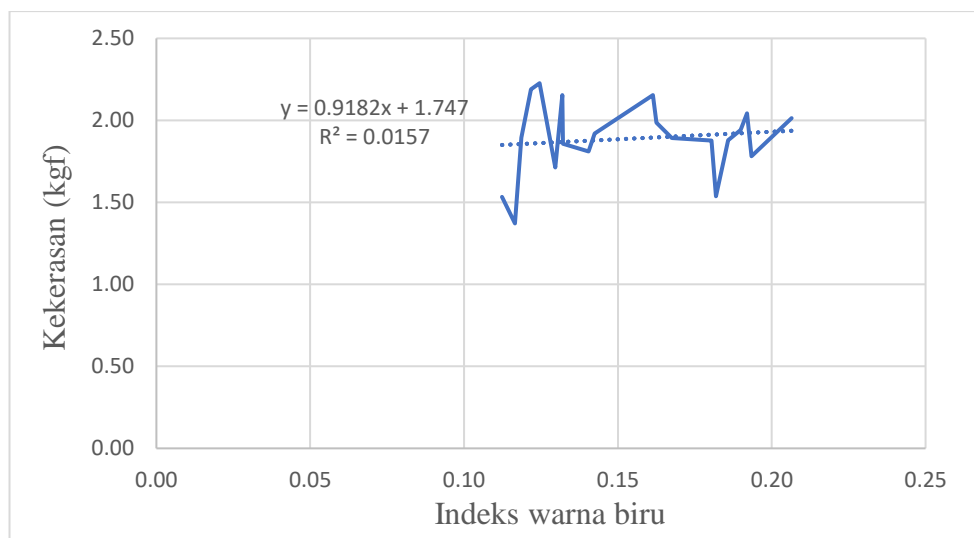
#### 4.5 Korelasi Nilai Indeks Warna dengan TPT Buah Jeruk



Gambar 9. Hubungan nilai *red* terhadap nilai kekerasan buah jeruk



Gambar 10. Hubungan nilai *green* terhadap nilai kekerasan buah jeruk



Gambar 11. Hubungan nilai *blue* terhadap nilai kekerasan buah jeruk

Nilai indeks warna RGB dari citra buah jeruk dianalisis untuk menemukan korelasi dengan tingkat kekerasan buah, yang telah diukur sebelumnya di laboratorium dan divisualisasikan melalui grafik. Grafik tersebut menghasilkan persamaan linear untuk indeks warna merah, hijau, dan biru, beserta nilai akurasi masing-masing yang dinyatakan dalam *R-squared*. Berdasarkan Gambar 9, 10, dan 11, diperoleh nilai *R-squared* sebesar 0,0019 untuk indeks warna merah, 0,0012 untuk indeks warna hijau, dan 0,0157 untuk indeks warna biru.

Dari hasil ini, terlihat bahwa indeks warna biru memiliki tingkat akurasi tertinggi dalam prediksi tingkat kekerasan buah jeruk. Namun, nilai indeks warna biru memiliki *R-squared* yang tergolong rendah sehingga juga belum cukup memadai untuk digunakan dalam estimasi yang akurat mengenai tingkat kekerasan. Meski begitu, indeks warna biru tetap digunakan dalam model ini mengingat akurasinya yang lebih tinggi daripada ketiga indeks warna lainnya. Oleh karena itu, persamaan yang dihasilkan dari grafik korelasi antara indeks warna biru dan data tingkat kekerasan yang diukur di laboratorium dipilih sebagai model utama untuk estimasi tingkat kekerasan buah jeruk, mengingat akurasinya yang lebih tinggi.

## **BAB V. PENUTUP**

### **5.1 Simpulan**

Berdasarkan tugas akhir ini dapat disimpulkan bahwa diameter mayor, diameter minor, massa dan luas area buah jeruk memiliki hubungan berbanding lurus dengan jumlah piksel citra dengan nilai *R-squared* di atas 0.5 dan mendekati nilai 1. Namun, tingkat kemanisan buah jeruk dan tingkat kekerasan buah jeruk tidak dapat diduga dengan indeks warna merah, hijau, ataupun biru karena menghasilkan nilai *R-squared* yang mendekati nilai 0. Meskipun sebagian besar hubungan yang terbentuk dengan nilai jumlah piksel memiliki nilai *R-squared* yang mendekati 1 hasil pendugaan yang diperoleh masih dapat menimbulkan galat yang sedikit jauh. Hal tersebut dapat disebabkan oleh pengambilan citra ataupun proses pengolahan citra, seperti *thresholding* dan *masking* yang kurang baik. Selain itu, penggunaan nilai rata-rata dalam proses kalibrasi jumlah piksel dan nilai indeks warna juga dapat menjadi faktor kurangnya akurasi pendugaan citra pada proses *threshold* dan *masking* pada buah jeruk.

### **5.2 Saran**

Hasil pengolahan citra yang telah dilakukan sudah cukup mewakili kondisi sebenarnya meskipun masih memiliki galat. Namun, galat yang dihasilkan tidak terlalu jauh dari data hasil yang sebenarnya. Hal tersebut dapat disebabkan oleh data yang tidak konsisten dan proses kalibrasi yang masih belum sempurna sehingga hasil analisis pendugaan citra yang dilakukan oleh program masih belum maksimal. Oleh karena itu, untuk penelitian berikutnya, perlu adanya perbaikan pada metode pengolahan citra, baik dalam kode pemrograman atau proses pengambilan sampel citra untuk menunjang hasil pendugaan yang lebih akurat dan mewakili kondisi yang sebenarnya berdasarkan data hasil yang diperoleh melalui uji laboratorium.

## DAFTAR PUSTAKA

- Ahmad U. 2010. Aplikasi teknik pengolahan citra dalam analisis non-destruktif produk pangan. *Jurnal Pangan*. 19(1):71-80.
- Andono PN, Sutojo T. 2018. *Pengolahan citra digital*. Penerbit Andi.
- Arifin N, Insani CN, Rasyid MR. 2023. Klasifikasi tingkat kematangan buah tomat menggunakan *computer vision* untuk *smart agriculture*. *Jurnal SAINTIKOM (Jurnal Sains Manajemen Informatika dan Komputer)*. 22(2):509-516.
- Dijaya R, Setiawan H. 2023. *Buku Ajar Pengolahan Citra Digital*. Umsida Press.
- Hadipernata M, Somantri AS, Hayuningtyas M, Hidayah N, Hoerudin H. 2020. Sistem deteksi cepat mutu organoleptik beras berbasis android. *Jurnal Aplikasi Teknologi Pangan*. 9(4):167-174.
- Harani NH, Hasanah M. 2020. *Deteksi Objek dan Pengenalan Karakter Plat Nomor Kendaraan Indonesia Berbasis Python*. Bandung: Kreatif.
- Hari SSJ, Gunawan R, Binanto I, Kumalasanti RA, Widyastuti W. 2023. *Pengantar Pembelajaran Mesin Menggunakan Bahasa Pemrograman Python*. Sanata Dharma University Press.
- Mutaqin G, Kom S. 2023. *Teknik Penghapusan Kabut Pada Citra Digital*. Nas Media Pustaka.
- Nabijonovich SB, Najmaddin G. 2024. Optimizing pyqt5 development with qt designer. *Web of Teachers: Inderscience Research*. 2(4): 254-259.
- Rahma L, Syaputra H, Mirza AH, Purnamasari SD. 2021. Objek deteksi makanan khas palembang menggunakan algoritma YOLO (*you only look once*). *Jurnal Nasional Ilmu Komputer*. 2(3):213-232.
- Rahman S, Sembiring A, Siregar D, Prahmana IG, Puspadini R, Zen M. 2023. *Python: Dasar dan Pemrograman Berorientasi Objek*. Penerbit Tahta Media.
- Satriorini MGRF, Sumiharto R, Hujja RM. 2022. Sistem pemantauan pertumbuhan anggrek berdasarkan pengolahan citra digital. *IJEIS (Indonesian Journal of Electronics and Instrumentation Systems)*. 12(2): 203-212.
- Suhaeni N. 2023. *Petunjuk Praktis Menanam Jeruk*. Bandung: Nuansa Cendekia.



## LAMPIRAN

### Lampiran 1. Data Hasil Pengukuran Laboratorium Jeruk Mentah

Tabel 1 Dimensi dan massa jeruk mentah

Sampel Jeruk	Diameter (cm)		Luas (cm <sup>2</sup> )	Massa (g)
	Mayor	Minor		
M1	7.225	6.125	69.48	160.09
M2	7.225	6.220	70.56	190.75
M3	7.175	6.325	71.25	170.89
M4	6.900	6.150	66.62	166.66
M5	6.805	6.415	68.54	164.15
M6	6.905	5.975	64.77	156.77
M7	6.610	6.100	63.30	149.03
M8	6.870	6.425	69.30	161.32
M9	6.525	6.260	64.13	150.29
M10	6.780	6.000	63.87	154.89
<b>Rata-rata</b>	6.902	6.200	67.18	162.48

Tabel 2 Kekerasan jeruk mentah

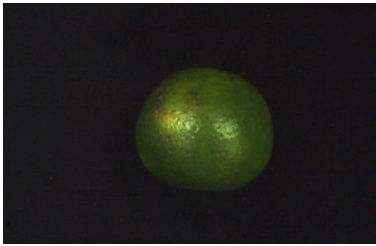
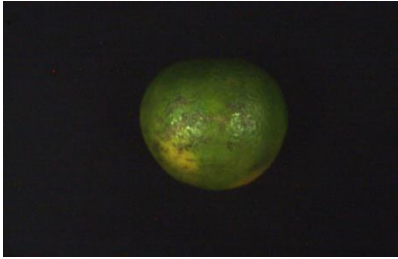



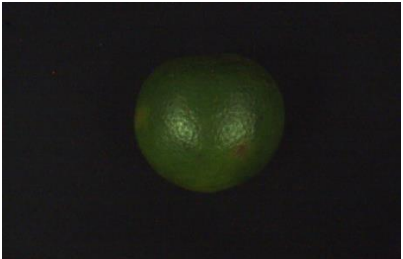
Sampel Jeruk	Kekerasan (kg)					
	Pangkal 1	Pangkal 2	Tengah 1	Tengah 2	Ujung 1	Ujung 2
M1	2.27	1.97	1.67	1.85	1.87	1.72
M2	2.08	2.26	1.95	2.05	1.57	2.01
M3	2.46	2.04	1.82	2.07	2.23	1.64
M4	2.14	2.46	2.54	2.25	1.90	1.64
M5	1.84	1.76	2.07	1.60	2.00	1.42
M6	1.75	2.14	1.53	2.03	1.73	2.08
M7	1.64	1.67	1.72	2.06	2.34	1.84
M8	2.56	2.03	2.01	1.48	1.59	2.41
M9	2.03	2.03	1.95	1.94	2.05	1.65
M10	1.66	2.11	1.42	0.86	1.65	1.52
<b>Rata-rata</b>	2.04	2.05	1.87	1.82	1.89	1.79


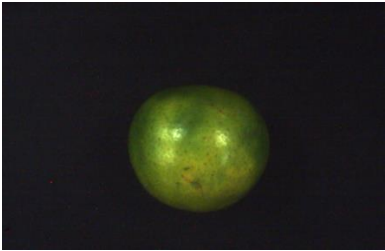

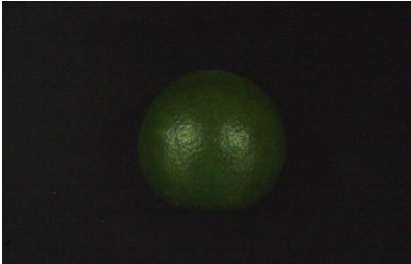
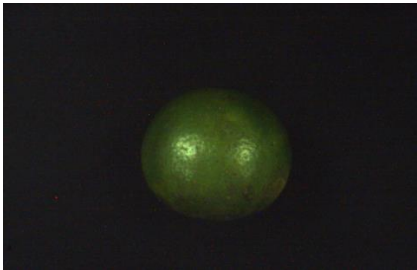
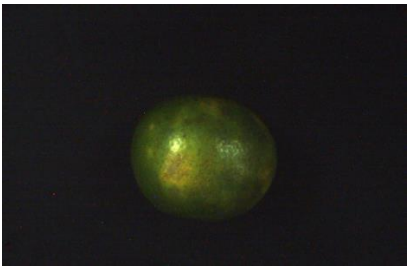
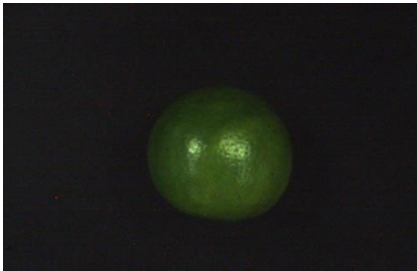
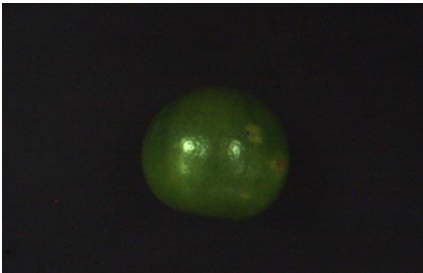
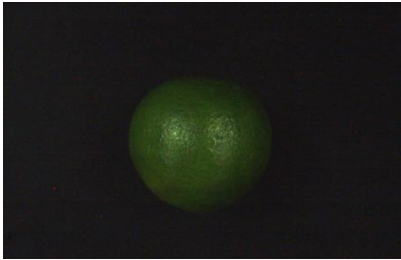

Tabel 3 TPT jeruk mentah

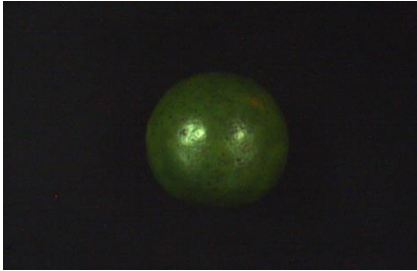
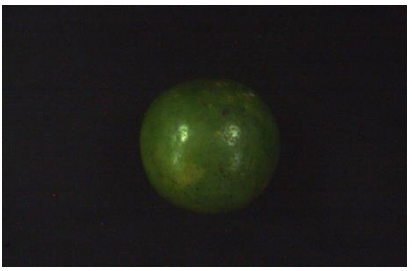
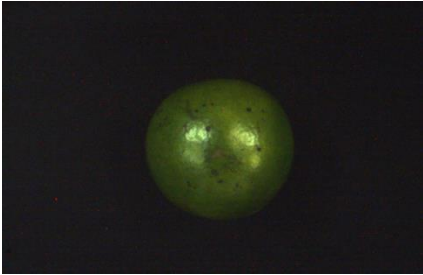

Sampel Jeruk	TPT (brix)			
	Ulangan 1	Ulangan 2	Ulangan 3	Rata-Rata
M1	8.40	8.50	8.50	8.47
M2	9.20	9.40	9.10	9.23
M3	7.80	7.80	7.90	7.83
M4	8.70	8.60	8.60	8.63

Sampel Jeruk	TPT (brix)			
	Ulangan 1	Ulangan 2	Ulangan 3	Rata-Rata
M5	8.90	8.90	9.00	8.93
M6	9.00	9.20	9.10	9.10
M7	9.50	9.50	9.50	9.50
M8	7.80	7.90	7.80	7.83
M9	9.10	9.10	9.20	9.13
M10	9.40	9.40	9.40	9.40
<b>Rata-rata</b>	8.78	8.83	8.81	8.81

Tabel 4 Hasil perekaman citra jeruk mentah

Sampel Jeruk	Gambar	
	Depan	Belakang
M1		
M2		
M3		

Sampel Jeruk	Gambar	
	Depan	Belakang
M4		
M5		
M6		
M7		
M8		

Sampel Jeruk	Gambar	
	Depan	Belakang
M9		
M10		

Lampiran 2. Data Hasil Pengukuran Laboratorium Jeruk Matang

Tabel 5 Dimensi dan massa jeruk matang

Sampel	Diameter (cm)		Luas (cm <sup>2</sup> )	Massa (g)
	Mayor	Minor		
J1	6.450	5.870	59.443	137.820
J2	7.050	6.580	72.831	189.140
J3	6.660	5.778	60.416	155.890
J4	6.242	5.062	49.607	122.240
J5	6.750	6.120	64.857	153.880
J6	5.962	5.580	52.231	122.450
J7	5.945	5.550	51.802	119.590
J8	6.654	6.040	63.099	154.860
J9	6.440	5.535	55.963	141.280
J10	5.790	5.310	48.269	108.810
<b>Rata-Rata</b>	6.394	5.743	57.649	140.596

Tabel 6 Kekerasan jeruk matang







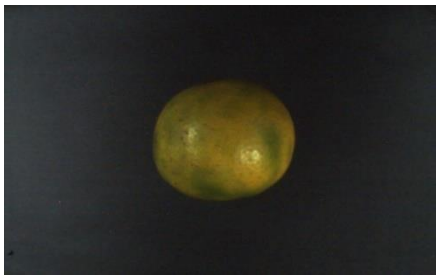

Sampel	Kekerasan (kg)					
	Pangkal 1	Pangkal 2	Tengah 1	Tengah 2	Ujung 1	Ujung 2
J1	2.01	1.91	2.86	2.09	2.40	2.09
J2	2.58	1.84	2.07	2.26	2.13	2.05
J3	2.31	2.00	2.26	2.08	2.01	2.48
J4	1.68	1.88	1.75	1.76	2.16	1.92
J5	1.83	1.74	1.78	1.61	2.00	1.90
J6	1.55	1.37	1.57	1.72	1.98	1.01
J7	1.26	1.12	1.50	1.22	1.76	1.37
J8	2.02	1.86	2.06	1.78	2.21	1.59
J9	1.75	1.80	2.03	1.66	1.55	1.49
J10	1.56	1.38	2.00	2.52	2.07	1.84
<b>Rata-Rata</b>	1.86	1.69	1.99	1.87	2.03	1.77







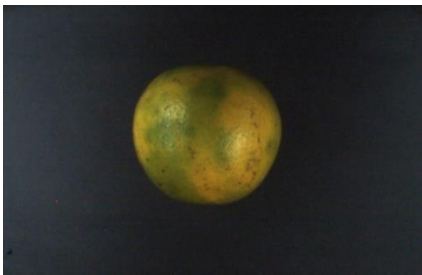



Tabel 7 TPT jeruk matang



Sampel	TPT (brix)			
	Ulangan 1	Ulangan 2	Ulangan 3	Rata-Rata
J1	8.20	8.10	8.40	8.23
J2	8.70	8.80	8.60	8.70
J3	8.70	8.70	8.80	8.73
J4	9.70	9.70	9.70	9.70
J5	9.60	9.40	9.70	9.57
J6	10.00	9.80	9.90	9.90
J7	10.00	9.80	9.80	9.87

Sampel	TPT (brix)			
	Ulangan 1	Ulangan 2	Ulangan 3	Rata-Rata
J8	9.10	9.20	9.10	9.13
J9	9.00	9.00	9.10	9.03
J10	7.80	7.90	7.90	7.87
<b>Rata-Rata</b>	9.08	9.04	9.10	9.07

Tabel 8 Hasil perekaman citra jeruk matang

Sampel	Gambar	
	Depan	Belakang
J1		
J2		
J3		
J4		

Sampel	Gambar	
	Depan	Belakang
J5		
J6		
J7		
J8		
J9		

Sampel	Gambar	
	Depan	Belakang
J10		



Lampiran 3. Data hasil pengolahan citra melalui GUI

Tabel 9 Hasil pengolahan citra (*RGB, pixel, HSI*)

Sampel	Nama File	Red	Green	Blue	Pixel Count	Hue	Saturation	Intensity
Mentah	M1 Belakang	0.39	0.45	0.16	55336	73.24 °	63.34%	44.91%
	M1 Depan	0.37	0.46	0.17	54677	77.79 °	61.99%	45.53%
	M2 Belakang	0.35	0.45	0.19	60632	83.65 °	57.01%	45.35%
	M2 Depan	0.41	0.45	0.13	61455	67.80 °	70.64%	45.42%
	M3 Belakang	0.35	0.45	0.20	56021	84.25 °	55.04%	44.90%
	M3 Depan	0.37	0.44	0.18	55324	76.98 °	58.47%	44.45%
	M4 Belakang	0.39	0.46	0.15	54161	73.53 °	68.68%	46.33%
	M4 Depan	0.36	0.46	0.18	54267	83.12 °	61.19%	46.46%
	M5 Belakang	0.33	0.45	0.21	53108	89.55 °	53.74%	45.50%
	M5 Depan	0.35	0.47	0.18	52680	84.75 °	61.97%	47.07%
	M6 Belakang	0.39	0.44	0.17	49154	71.21 °	61.38%	44.02%
	M6 Depan	0.36	0.45	0.19	49683	81.77 °	57.75%	45.19%
	M7 Belakang	0.34	0.47	0.18	48949	86.77 °	61.07%	47.24%
	M7 Depan	0.34	0.48	0.19	49271	88.87 °	60.72%	47.60%
	M8 Belakang	0.33	0.45	0.22	42934	89.49 °	51.92%	44.93%
	M8 Depan	0.34	0.47	0.19	50649	89.17 °	58.75%	47.02%
	M9 Belakang	0.36	0.46	0.19	50221	82.73 °	59.27%	45.81%
	M9 Depan	0.35	0.46	0.19	48774	84.63 °	57.75%	45.76%
	M10 Belakang	0.35	0.46	0.19	51831	83.05 °	58.27%	45.59%
	M10 Depan	0.37	0.46	0.17	50727	78.36 °	62.19%	45.71%
Matang	J1 Belakang	0.51	0.36	0.13	48454	37.03 °	74.56%	50.79%
	J1 Depan	0.51	0.37	0.12	48164	38.08 °	76.30%	51.07%
	J2 Belakang	0.47	0.40	0.13	61424	48.18 °	72.59%	46.93%
	J2 Depan	0.46	0.41	0.14	61430	50.05 °	70.53%	45.92%
	J3 Belakang	0.50	0.38	0.12	51783	41.53 °	76.36%	49.97%
	J3 Depan	0.50	0.38	0.13	51794	41.00 °	74.74%	49.61%
	J4 Belakang	0.47	0.39	0.14	44700	46.63 °	70.77%	46.85%
	J4 Depan	0.47	0.40	0.13	43030	48.41 °	72.83%	46.92%
	J5 Belakang	0.44	0.42	0.14	52318	57.11 °	67.77%	43.67%
	J5 Depan	0.46	0.40	0.14	51589	48.32 °	69.82%	46.17%
	J6 Belakang	0.48	0.40	0.11	42324	47.17 °	76.53%	48.28%
	J6 Depan	0.48	0.41	0.11	42699	47.33 °	77.09%	48.39%
	J7 Belakang	0.49	0.40	0.11	41248	45.81 °	76.80%	48.77%
	J7 Depan	0.48	0.40	0.12	41465	46.24 °	75.01%	48.13%

Sampel	Nama File	Red	Green	Blue	Pixel Count	Hue	Saturation	Intensity
	J8 Belakang	0.43	0.41	0.15	52601	56.04 °	64.28%	43.20%
	J8 Depan	0.46	0.40	0.13	53665	49.32 °	71.63%	46.38%
	J9 Belakang	0.46	0.40	0.14	48241	48.81 °	70.34%	46.18%
	J9 Depan	0.48	0.39	0.12	48103	44.90 °	74.91%	48.49%
	J10 Belakang	0.47	0.40	0.12	39089	48.42 °	73.61%	47.13%
	J10 Depan	0.49	0.40	0.11	38431	45.31 °	76.74%	48.91%

Tabel 10 Hasil pengolahan citra (dimensi, *brix*, kekerasan)

Sampel	Nama File	Mayor (cm)	Minor (cm)	Luas (cm <sup>2</sup> )	Massa (gram)	Brix (%)	Kekerasan (kgf)
Mentah	M1 Belakang	6.69	6.06	65.68	167.53	9.14%	1.96
	M1 Depan	6.65	6.03	64.95	165.36	9.31%	2.01
	M2 Belakang	7.00	6.33	71.50	185.01	9.75%	2.16
	M2 Depan	7.05	6.37	72.41	187.73	8.51%	1.75
	M3 Belakang	6.73	6.10	66.43	169.79	9.89%	2.20
	M3 Depan	6.69	6.06	65.67	167.49	9.54%	2.09
	M4 Belakang	6.62	6.00	64.39	163.66	8.75%	1.83
	M4 Depan	6.62	6.01	64.50	164.01	9.46%	2.06
	M5 Belakang	6.55	5.95	63.23	160.18	10.06%	2.26
	M5 Depan	6.53	5.93	62.76	158.77	9.43%	2.05
	M6 Belakang	6.32	5.75	58.88	147.13	9.25%	1.99
	M6 Depan	6.35	5.78	59.46	148.88	9.67%	2.13
	M7 Belakang	6.30	5.74	58.65	146.46	9.53%	2.08
	M7 Depan	6.32	5.76	59.01	147.52	9.59%	2.11
	M8 Belakang	5.94	5.44	52.04	126.61	10.17%	2.30
	M8 Depan	6.41	5.83	60.52	152.07	9.73%	2.15
	M9 Belakang	6.38	5.81	60.05	150.65	9.58%	2.10
	M9 Depan	6.29	5.73	58.46	145.88	9.72%	2.15
	M10 Belakang	6.48	5.89	61.82	155.97	9.65%	2.13
	M10 Depan	6.41	5.83	60.61	152.32	9.30%	2.01
Matang	J1 Belakang	6.27	5.72	58.11	144.82	8.43%	1.72
	J1 Depan	6.26	5.70	57.79	143.87	8.27%	1.66
	J2 Belakang	7.05	6.37	72.38	187.62	8.42%	1.72
	J2 Depan	7.05	6.37	72.38	187.64	8.55%	1.76
	J3 Belakang	6.47	5.88	61.77	155.81	8.21%	1.65

<b>Sampel</b>	<b>Nama File</b>	<b>Mayor (cm)</b>	<b>Minor (cm)</b>	<b>Luas (cm<sup>2</sup>)</b>	<b>Massa (gram)</b>	<b>Brix (%)</b>	<b>Kekerasan (kgf)</b>
	J3 Depan	6.47	5.88	61.78	155.84	8.35%	1.69
	J4 Belakang	6.05	5.53	53.98	132.43	8.59%	1.77
	J4 Depan	5.95	5.45	52.14	126.92	8.40%	1.71
	J5 Belakang	6.51	5.91	62.36	157.57	8.66%	1.80
	J5 Depan	6.46	5.87	61.56	155.17	8.63%	1.79
	J6 Belakang	5.91	5.41	51.37	124.59	8.11%	1.61
	J6 Depan	5.93	5.43	51.78	125.83	8.06%	1.60
	J7 Belakang	5.84	5.36	50.18	121.04	8.11%	1.61
	J7 Depan	5.85	5.37	50.42	121.76	8.25%	1.66
	J8 Belakang	6.52	5.92	62.67	158.51	8.93%	1.89
	J8 Depan	6.59	5.98	63.84	162.02	8.48%	1.74
	J9 Belakang	6.26	5.71	57.87	144.12	8.59%	1.77
	J9 Depan	6.25	5.70	57.72	143.66	8.28%	1.67
	J10 Belakang	5.71	5.25	47.81	113.92	8.33%	1.69
	J10 Depan	5.67	5.22	47.08	111.75	8.12%	1.62

Lampiran 4. Dokumentasi Pengambilan data laboratorium



Gambar 12. Pelabelan sampel buah jeruk



Gambar 13. Pencucian sampel buah jeruk



Gambar 14. Proses perekaman citra pada sampel



Gambar 15. Proses pengukuran dimensi sampel buah jeruk



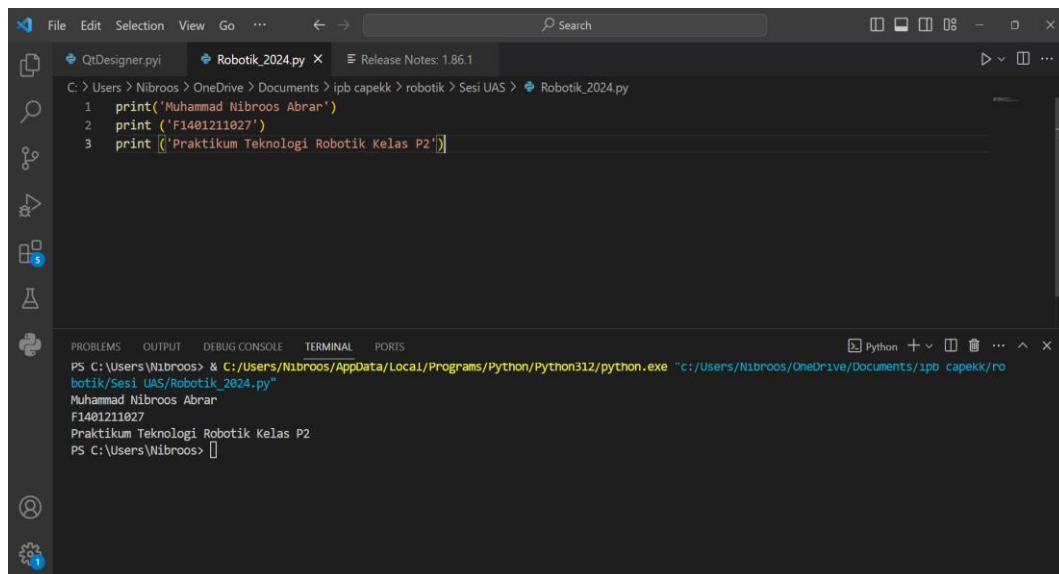


Gambar 16. Proses pengukuran kekerasan sampel buah jeruk

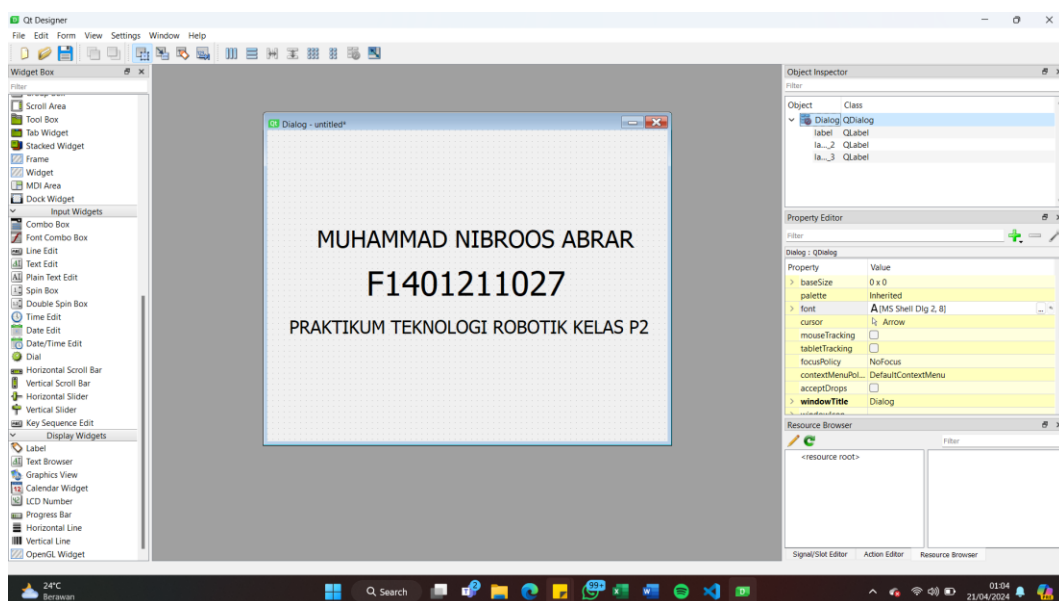


Gambar 17. Proses pengukuran massa sampel buah jeruk

## Lampiran 5. Tampilan aplikasi Visual Studio Code dan *Qt Designer*



Gambar 18. Aplikasi *Visual Studio Code*

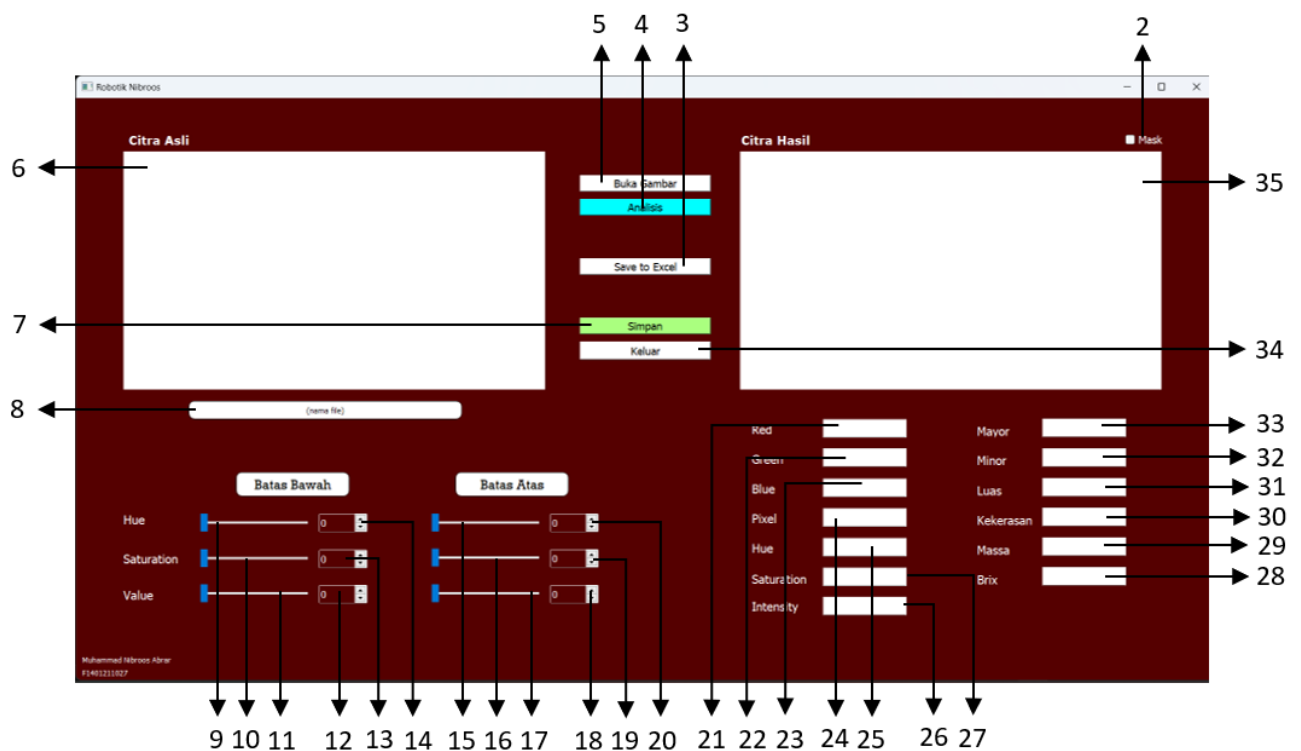


Gambar 19. Aplikasi *Qt Designer*

## Lampiran 6. Tampilan GUI Pengolahan Citra



Gambar 20. Tampilan awal GUI



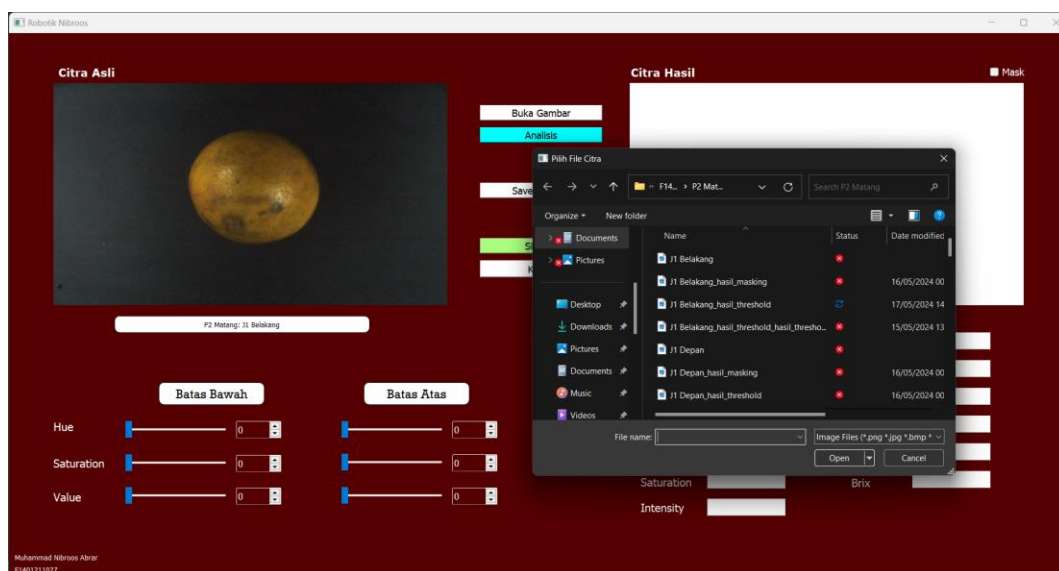
Gambar 21. Tampilan analisis citra GUI



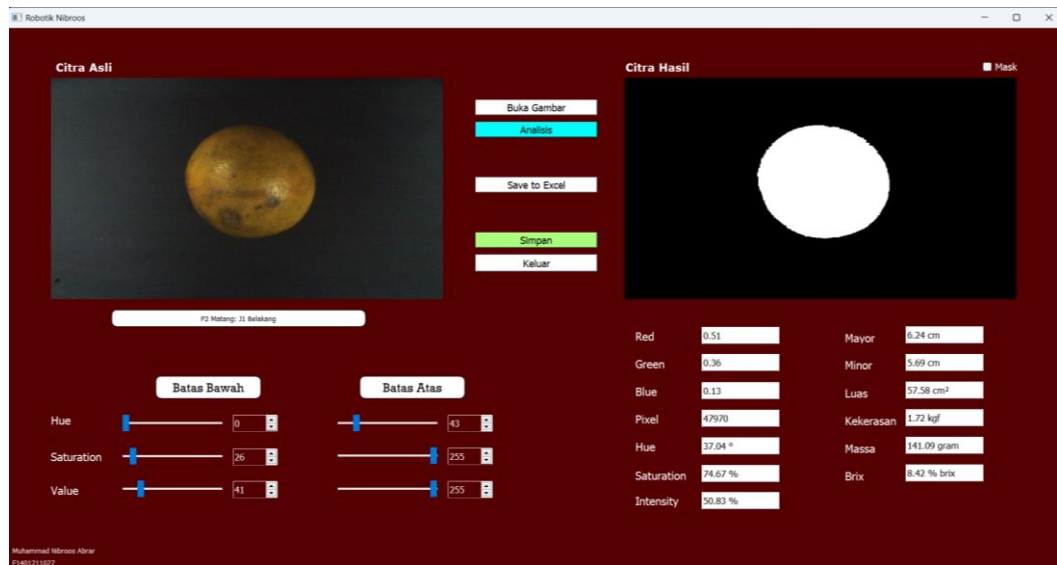
Tabel 11. Keterangan nama dan fungsi *object* pada GUI

No	Nama Objek	Fungsi Object
1	start	Menampilkan halaman analisis citra
2	mask_check	Menampilkan hasil <i>masking</i> citra
3	excel	Menyimpan hasil analisis ke <i>file</i> Microsoft Excel
4	analisis	Menampilkan hasil kuantitatif pengolahan citra
5	buka	Membuka <i>file</i> citra yang akan dianalisis
6	asal	Tempat untuk menampilkan file citra yang akan dianalisis
7	simpan	Menyimpan hasil <i>threshold</i> dan <i>masking</i> citra
8	nama_file	Menampilkan nama <i>file</i> citra yang akan dianalisis
9	hmin	<i>Slider</i> untuk <i>hue</i> batas bawah
10	smin	<i>Slider</i> untuk saturasi batas bawah
11	vmin	<i>Slider</i> untuk <i>value</i> batas bawah
12	vmin_spin	Kotak keluaran untuk menunjukkan nilai <i>value</i> batas bawah
13	smin_spin	Kotak keluaran untuk menunjukkan nilai saturasi batas bawah
14	hmin_spin	Kotak keluaran untuk menunjukkan nilai <i>hue</i> batas bawah
15	hmax	<i>Slider</i> untuk <i>hue</i> batas atas
16	smax	<i>Slider</i> untuk saturasi batas atas
17	vmax	<i>Slider</i> untuk <i>value</i> batas atas
18	vmax_spin	Kotak keluaran untuk menunjukkan nilai <i>value</i> batas atas
19	smax_spin	Kotak keluaran untuk menunjukkan nilai saturasi batas atas
20	hmax_spin	Kotak keluaran untuk menunjukkan nilai <i>hue</i> batas atas
21	red_val	Menampilkan nilai <i>Red</i> pada citra yang dianalisis
22	green_val	Menampilkan nilai <i>Green</i> pada citra yang dianalisis

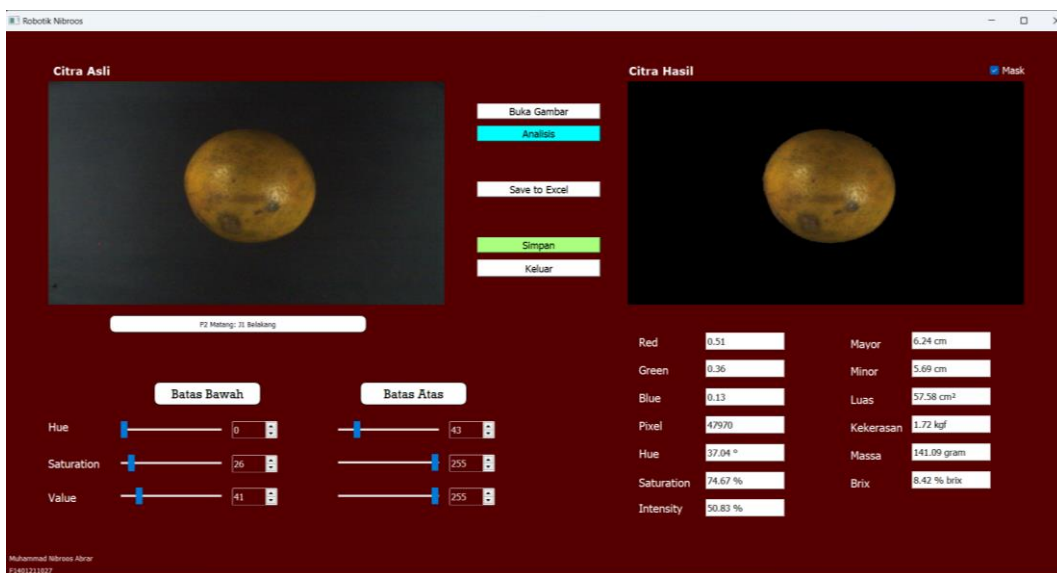
No	Nama Objek	Fungsi Object
23	blue_val	Menampilkan nilai <i>Blue</i> pada citra yang dianalisis
24	pixel_val	Menampilkan nilai <i>Pixel</i> pada citra yang dianalisis
25	hue_val	Menampilkan nilai <i>Hue</i> pada citra yang dianalisis
26	intens_val	Menampilkan nilai Intensitas pada citra yang dianalisis
27	sat_val	Menampilkan nilai Saturasi pada citra yang dianalisis
28	brix_val	Menampilkan nilai <i>Brix</i> pada citra yang dianalisis
29	massa_val	Menampilkan nilai Massa pada citra yang dianalisis
30	kekerasan_val	Menampilkan nilai Kekerasan pada citra yang dianalisis
31	luas_val	Menampilkan nilai Luas pada citra yang dianalisis
32	minor_val	Menampilkan nilai Minor pada citra yang dianalisis
33	mayor_val	Menampilkan nilai Mayor pada citra yang dianalisis
34	keluar	Mengakhiri program dan keluar dari tampilan GUI
35	hasil	Tempat untuk menampilkan citra hasil <i>threshold</i> atau hasil <i>masking</i>



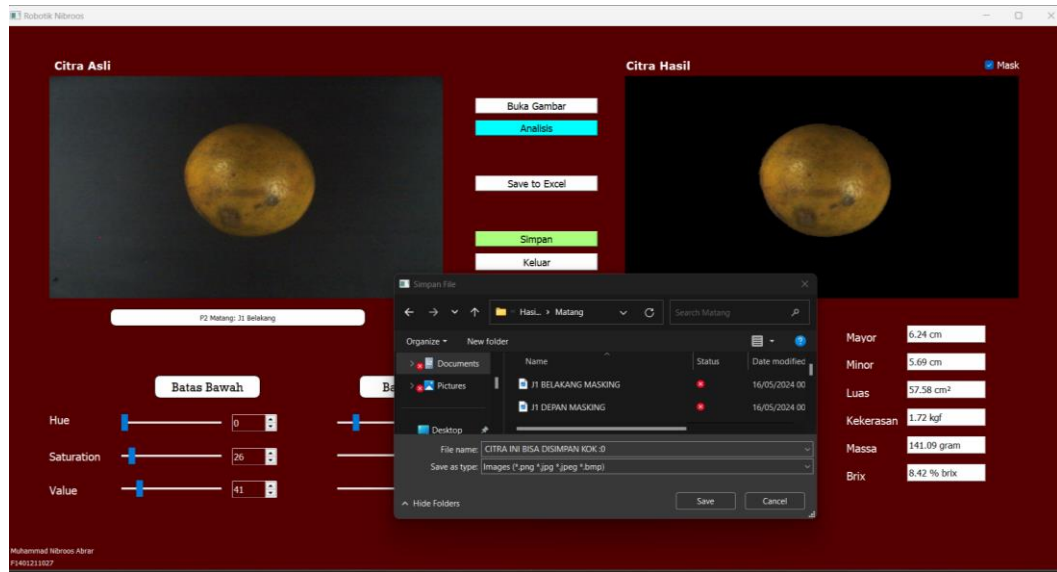
Gambar 22. Membuka *file* citra asal



Gambar 23. Segmentasi citra dengan luaran *threshold*

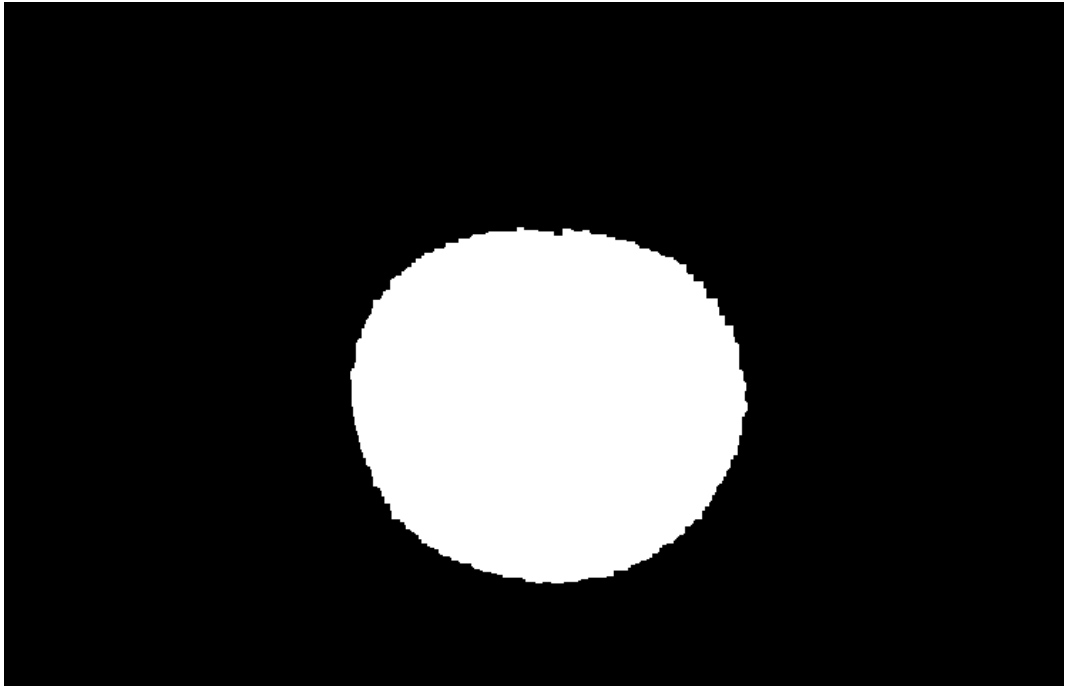


Gambar 24. Segmentasi citra dengan luaran *masking*

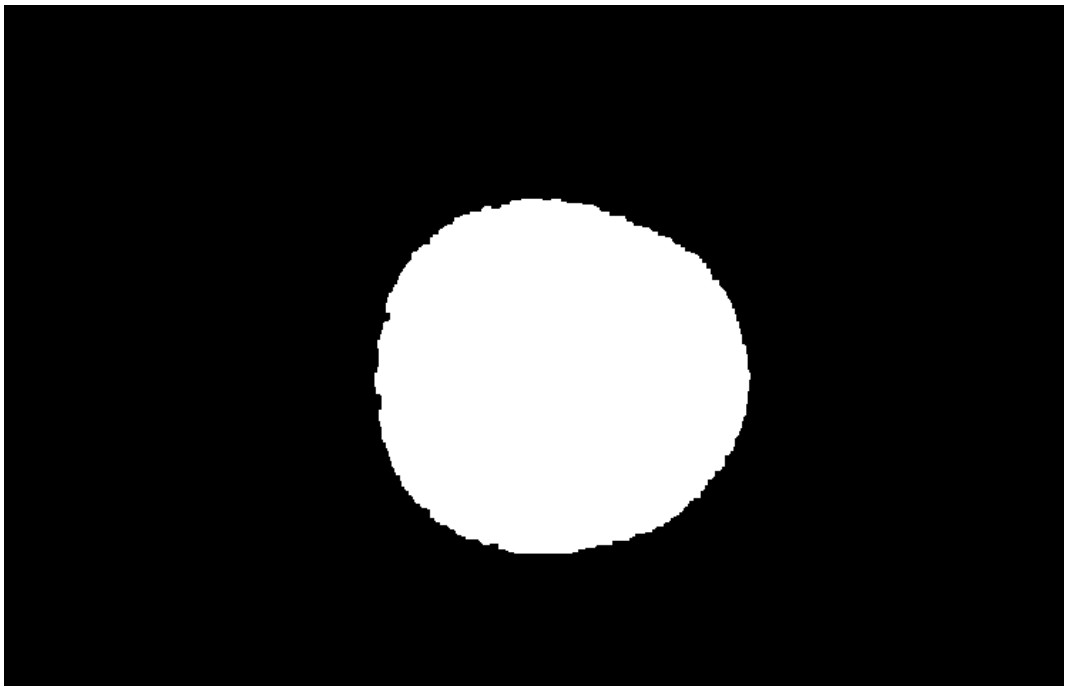


Gambar 25. Menyimpan *file* citra hasil

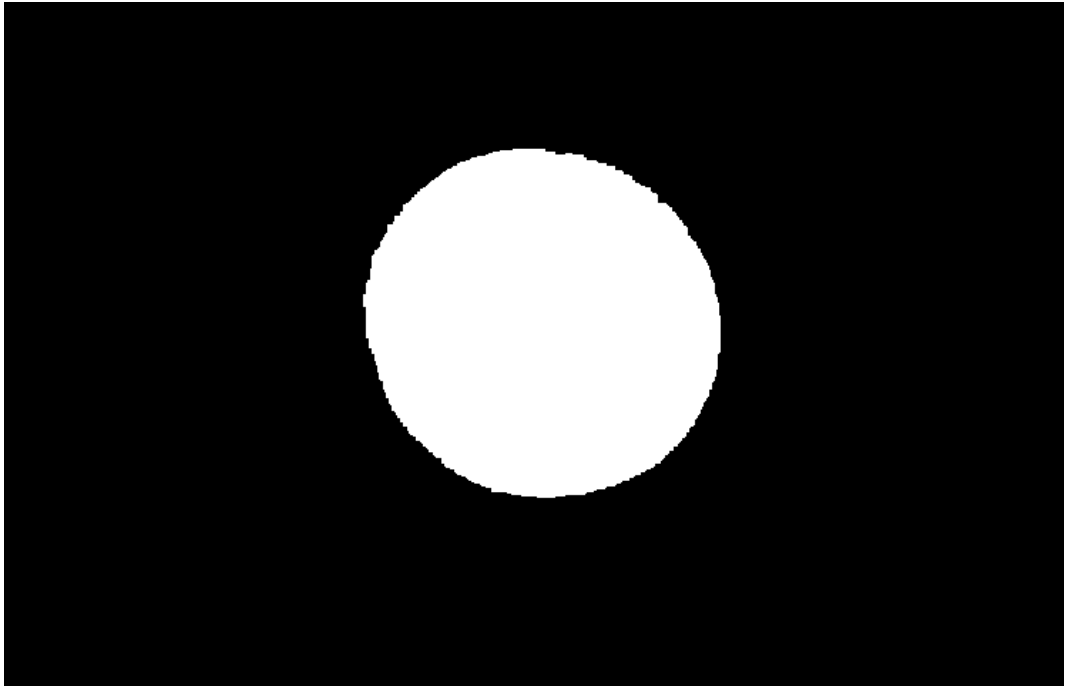
Lampiran 7. Citra Hasil *Threshold*



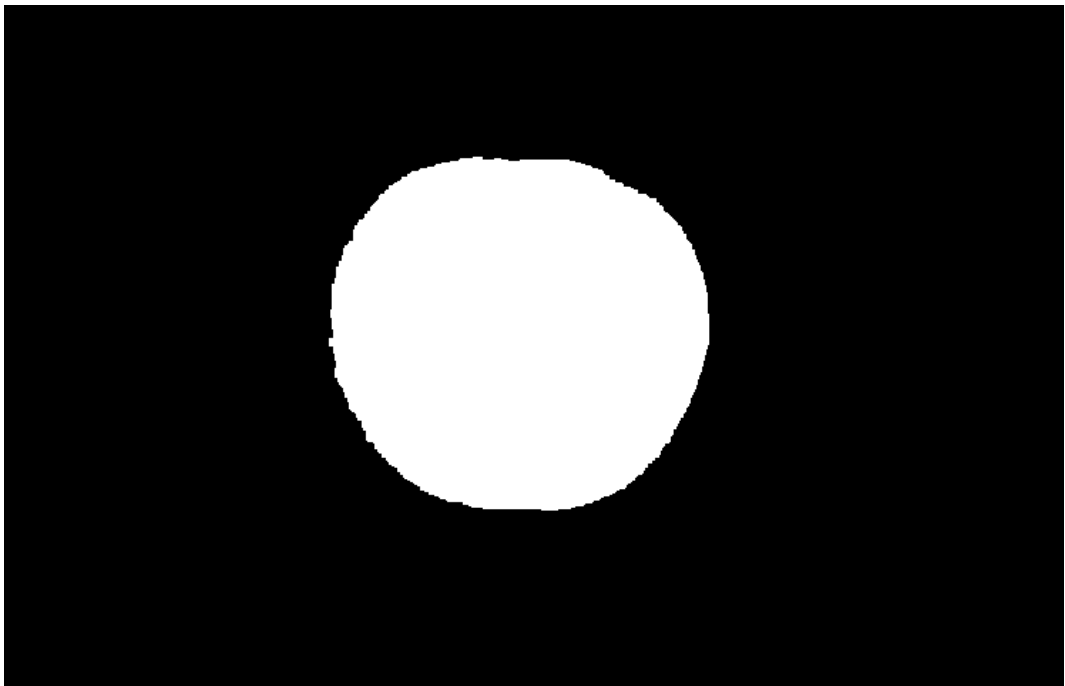
Gambar 26. Citra hasil *threshold* mentah 4A



Gambar 27. Citra hasil *threshold* mentah 10A

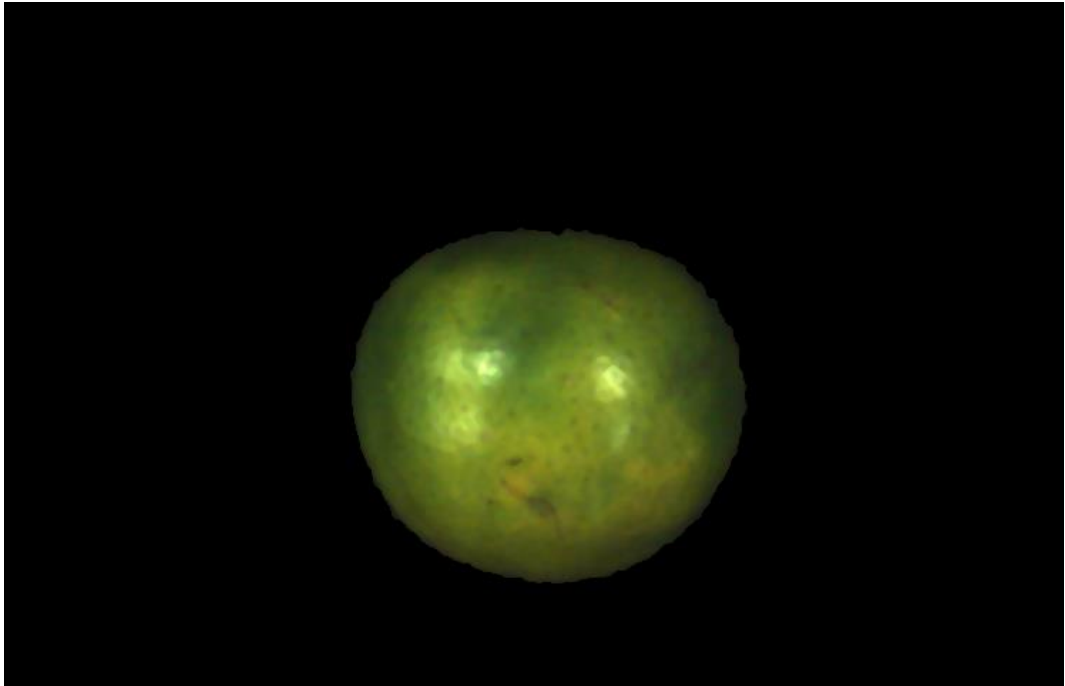


Gambar 28. Citra hasil *threshold* matang 1a

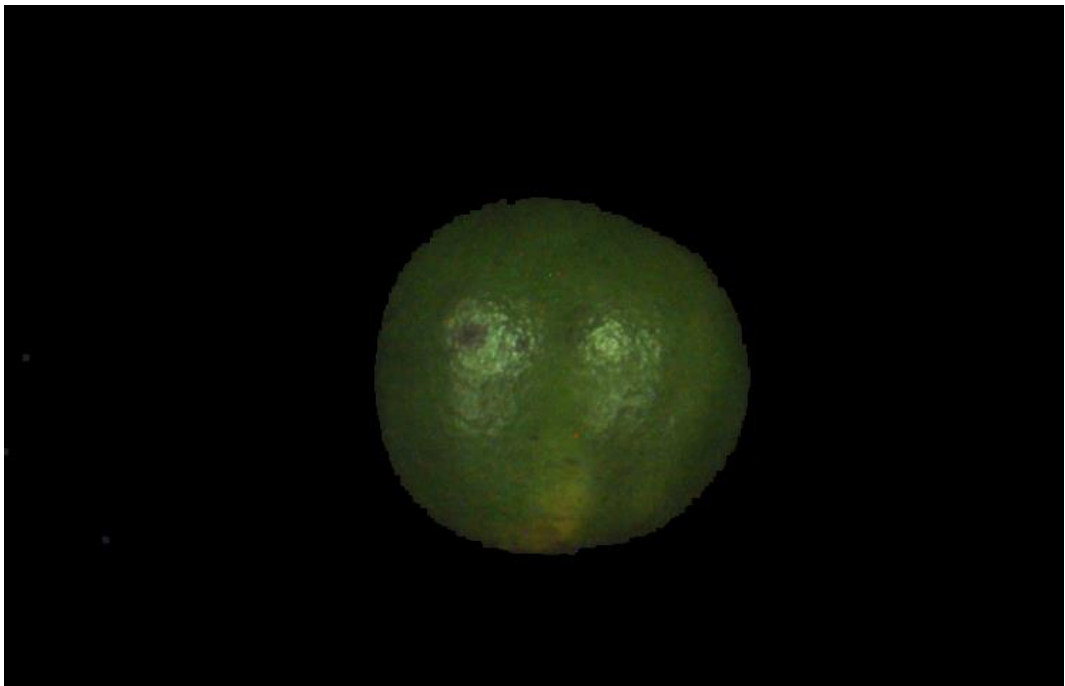


Gambar 29. Citra hasil *threshold* matang 8b

Lampiran 8. Citra Hasil *Masking*



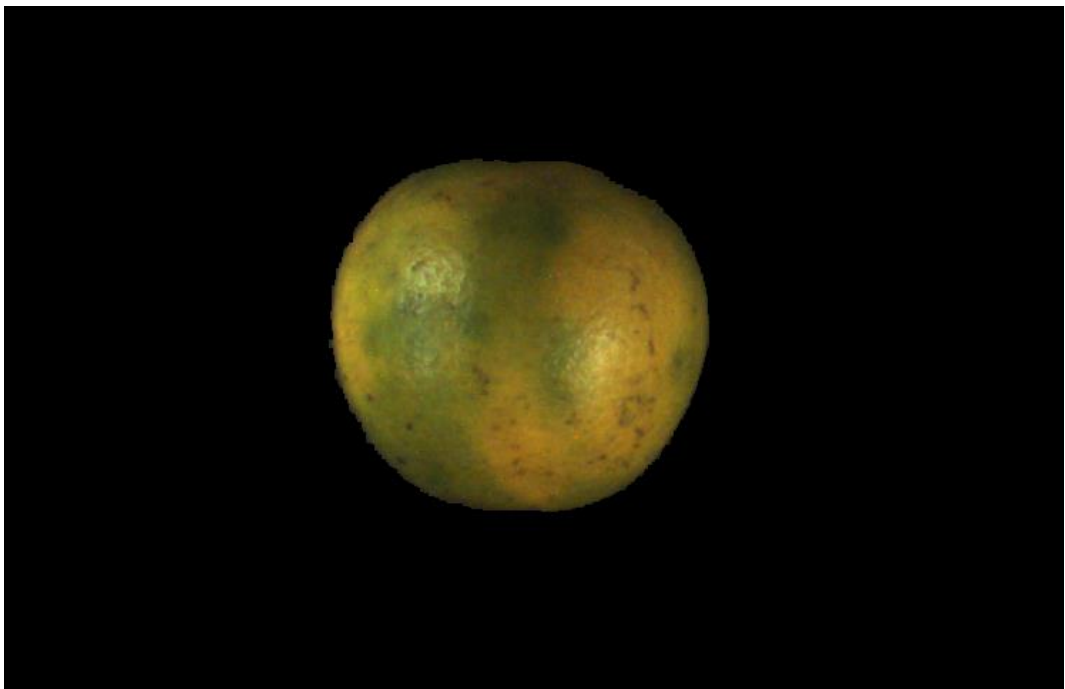
Gambar 30. Citra hasil *masking* mentah 4A



Gambar 31. Citra hasil *masking* mentah 10A



Gambar 32. Citra hasil *masking* matang 1a



Gambar 33. Citra hasil *masking* matang 8b



## Lampiran 9. Kode Pemrograman

```
# Diprogram oleh : Muhammad Nibroos Abrar - F1401211027
# Departemen Teknik Mesin dan Biosistem, FATETA-IPB University

import sys
import cv2
import numpy as np
import pathlib
from PyQt5 import QtCore, QtWidgets
from PyQt5.QtWidgets import QDialog, QDesktopWidget, QMainWindow,
QWidget, QFileDialog, QApplication, QLineEdit, QLabel, QCheckBox,
QComboBox, QHBoxLayout, QVBoxLayout, QPushButton, QSlider,
QGridLayout, QMessageBox, QAction
from PyQt5.uic import loadUi
from datetime import datetime
import csv
import os
import openpyxl
import pyautogui
from PyQt5 import QtGui
from PyQt5.QtCore import pyqtSlot
from PyQt5.QtGui import QImage, QPixmap
from numpy.lib.twodim_base import mask_indices
# from object_detector import *

class Robotik(QDialog):
    def __init__(self):
        super(Robotik, self).__init__()
        loadUi("./tampilanawal.ui", self)

        self.start.clicked.connect(self.gotodashboard) # Mengaktifkan button start
        untuk menuju halaman kedua
        self.setWindowTitle('ROBOTIK NIBROOS') #menampilkan judul pada
        window GUI

        # Menampilkan halaman kedua
        def gotodashboard(self):
            halamanutama = MainWindow()
            widget.addWidget(halamanutama)
            widget.setCurrentIndex(widget.currentIndex() + 1)

class MainWindow(QDialog):
    def __init__(self):
        super(MainWindow, self).__init__()
        loadUi("./robotik.ui", self) #membaca ui dari qtdesigner
```

```

self.buka.clicked.connect(self.uploadfoto) # Mengaktifkan button buka
gambar
self.keluar.clicked.connect(self.Keluar) # Mengaktifkan button keluar
self.analisis.clicked.connect(self.Analys) # Mengaktifkan button analisis
self.simpan.clicked.connect(self.Save) # Mengaktifkan button simpan
self.mask_check.stateChanged.connect(self.Analys) #Mengaktifkan fungsi
checkbox
self.excel.clicked.connect(self.SaveToExcel) # Mengaktifkan fungsi tombol
save to excel
self.result_image = None #Menginisiasi kondisi awal Qlabel hasil ketika
belum klik 'analisis'
self.setWindowTitle('ROBOTIK NIBROOS') #memberi headline pada
tampilan gui

# Membuat workbook Excel
self.workbook = openpyxl.load_workbook("./datamentahRobotik.xlsx")
self.sheet = self.workbook.active
self.row_counter = 1

# Mengatur nilai minimum dan maksimum untuk slider dan spinbox
self.hmax.setMinimum(0)
self.hmax.setMaximum(255)
self.smax.setMinimum(0)
self.smax.setMaximum(255)
self.vmax.setMinimum(0)
self.vmax.setMaximum(255)
self.hmin.setMinimum(0)
self.hmin.setMaximum(255)
self.smin.setMinimum(0)
self.smin.setMaximum(255)
self.vmin.setMinimum(0)
self.vmin.setMaximum(255)

# Mengaktifkan slider
self.hmax.sliderMoved.connect(self.update_hmax_spin)
self.smax.sliderMoved.connect(self.update_smax_spin)
self.vmax.sliderMoved.connect(self.update_vmax_spin)
self.hmin.sliderMoved.connect(self.update_hmin_spin)
self.smin.sliderMoved.connect(self.update_smin_spin)
self.vmin.sliderMoved.connect(self.update_vmin_spin)

# mengaktifkan Spinbox
self.hmax_spin.valueChanged.connect(self.update_hmax_slider)
self.smax_spin.valueChanged.connect(self.update_smax_slider)
self.vmax_spin.valueChanged.connect(self.update_vmax_slider)
self.hmin_spin.valueChanged.connect(self.update_hmin_slider)

```

```

self.smin_spin.valueChanged.connect(self.update_smin_slider)
self.vmin_spin.valueChanged.connect(self.update_vmin_slider)

# Inisialisasi nilai slider dan spinbox
self.update_hmax_spin(self.hmax.sliderPosition())
self.update_smax_spin(self.smax.sliderPosition())
self.update_vmax_spin(self.vmax.sliderPosition())
self.update_hmin_spin(self.hmin.sliderPosition())
self.update_smin_spin(self.smin.sliderPosition())
self.update_vmin_spin(self.vmin.sliderPosition())

# Membatasi nilai minimum dan maksimum spinbox serta menyelaraskannya
dengan slider
def update_hmax_spin(self, value):
    self.hmax_spin.setValue(value)
    self.hmax_spin.setMaximum(value)

def update_smax_spin(self, value):
    self.smax_spin.setValue(value)
    self.smax_spin.setMaximum(value)

def update_vmax_spin(self, value):
    self.vmax_spin.setValue(value)
    self.vmax_spin.setMaximum(value)

def update_hmin_spin(self, value):
    self.hmin_spin.setValue(value)
    self.hmin_spin.setMinimum(value)

def update_smin_spin(self, value):
    self.smin_spin.setValue(value)
    self.smin_spin.setMinimum(value)

def update_vmin_spin(self, value):
    self.vmin_spin.setValue(value)
    self.vmin_spin.setMinimum(value)

def update_hmax_slider(self, value):
    self.hmax.setValue(value)
    self.hmax_spin.setMaximum(value)

def update_smax_slider(self, value):
    self.smax.setValue(value)
    self.smax_spin.setMaximum(value)

def update_vmax_slider(self, value):

```

```

self.vmax.setValue(value)
self.vmax_spin.setMaximum(value)

def update_hmin_slider(self, value):
    self.hmin.setValue(value)
    self.hmin_spin.setMinimum(value)

def update_smin_slider(self, value):
    self.smin.setValue(value)
    self.smin_spin.setMinimum(value)

def update_vmin_slider(self, value):
    self.vmin.setValue(value)
    self.vmin_spin.setMinimum(value)

# Membuka File Foto
def uploadfoto(self):
    self.img_name = QFileDialog.getOpenFileName(self, 'Pilih File Citra', "",
"Image Files (*.png *.jpg *.bmp *.jpeg);;All Files ()", "")[0]
    self.img = cv2.imread(self.img_name, 1)

    #self.ImageInsertedCheck = self.asal.pixmap()
    Pathimg_name = pathlib.Path(self.img_name)
    self.PictureName = Pathimg_name.stem #Membaca nama file
    self.GroupName = self.img_name.split('/')[2] # Membaca nama folder
    self.nama_file.setText(f'{self.GroupName}: {self.PictureName}') #
Menampilkan nama folder dan nama file pada QLabel
    self.label_NamaFile.setText(f'{self.GroupName} {self.PictureName}')

    #self.img_shape = img.shape
    open_path = self.img_name.replace(".png",
"_hasil_threshold.png").replace(".jpg", "_hasil_threshold.jpg").replace(".bmp",
"_hasil_threshold.bmp")
    cv2.imwrite(open_path, self.img) #mengatur agar file dapat dibuka sesuai
dengan lokasi file yang dipilih
    pixmap_source = QPixmap(open_path)
    self.asal.setPixmap(pixmap_source)
    self.asal.setScaledContents(True) # Mengatur agar gambar ditampilkan
sesuai dengan ukuran QLabel pada gui

    self.hsv = cv2.cvtColor(self.img, cv2.COLOR_BGR2HSV) #mengonversi
citra rgb menjadi hsv
    self.gray = cv2.cvtColor(self.img, cv2.COLOR_BGR2GRAY) #mengonversi
citra rgb menjadi gray

    # menghubungkan slider terhadap hasil pengolahan citra

```

```

self.hmax.sliderMoved.connect(self.Analys)
self.smax.sliderMoved.connect(self.Analys)
self.vmax.sliderMoved.connect(self.Analys)
self.hmin.sliderMoved.connect(self.Analys)
self.smin.sliderMoved.connect(self.Analys)
self.vmin.sliderMoved.connect(self.Analys)

# Proses Analisis
def Analys(self):
    hMin = sMin = vMin = hMax = sMax = vMax = 0
    #hMin = psMin = pvMin = phMax = psMax = pvMax = 0
    #img = cv2.imread('./Citra_Asal.jpg', 1)
    #output = img

    # get current positions of all trackbars
    hMin = float(self.hmin.sliderPosition())
    sMin = float(self.smin.sliderPosition())
    vMin = float(self.vmin.sliderPosition())
    hMax = float(self.hmax.sliderPosition())
    sMax = float(self.smax.sliderPosition())
    vMax = float(self.vmax.sliderPosition())

    #Set minimum and max HSV values to display
    lower = np.array([hMin, sMin, vMin])
    upper = np.array([hMax, sMax, vMax])
    mask = cv2.inRange(self.hsv, lower, upper)

    # Apply morphological operations to improve the binary image
    kernel = np.ones((5,5), np.uint8)
    secondTresh = cv2.morphologyEx(mask, cv2.MORPH_OPEN, kernel)
    finalthresh = cv2.morphologyEx(secondTresh, cv2.MORPH_CLOSE,
kernel)

    res = cv2.bitwise_and(self.img, self.img, mask=finalthresh)
    # res = cv2.dilate(mask,kernel,iterations = 1)
    # # Erosi mask dengan kernel berukuran 3x3
    # res = cv2.erode(mask,kernel,iterations = 1)
    # # Smoothing citra dengan Gaussian Blur
    # res = cv2.GaussianBlur(self.img,(5,5),0)
    # res = cv2.medianBlur(res, 5)

    # Menghitung masing-masing komponen nilai RGB
    self.count = cv2.countNonZero(finalthresh) # Menghitung warna piksel
    sum_color_per_row = np.sum(res, axis=0)
    sum_color = np.sum(sum_color_per_row, axis=0)
    totalColor = sum_color/self.count

```

```

r,g,b = self.hitungRGB(totalColor)
H,S,I = self.hitungHSV(r,g,b)
tinggi_cm,lebar_cm,area,masa,V,keras = self.hitungDimensiCitra(g)

#format angka desimal
# R = round(r,2)
# G = round(g,2)
# B = round(b,2)
R = (f'{r:.2f}')
G = (f'{g:.2f}')
B = (f'{b:.2f}')
Hue = (f'{H:.2f} \u00b0')
Sat = (f'{S:.2f} %')
Intensity = (f'{I:.2f} %')

# Konversi nilai desimal red, green, blue ke rentang 0-255

# red_val_int = convert_to_255(r)
# green_val_int = convert_to_255(g)
# blue_val_int = convert_to_255(b)

# self.red_val.setText(str(red_val_int))
# self.green_val.setText(str(green_val_int))
# self.blue_val.setText(str(blue_val_int))

self.red_val.setText(str(R))
self.green_val.setText(str(G))
self.blue_val.setText(str(B))

self.pixel_val.setText(str(self.count))
self.hue_val.setText(str(Hue))
self.sat_val.setText(str(Sat))
self.intens_val.setText(str(Intensity))
self.mayor_val.setText(str(tinggi_cm))
self.minor_val.setText(str(lebar_cm))
self.luas_val.setText(str(area))
self.massa_val.setText(str(masa))
self.brix_val.setText(str(V))
self.kekerasan_val.setText(str(keras))

# Create HSV Image and threshold into a range.
#output = cv2.bitwise_and(img,img, mask= mask)

# Simpan hasil thresholding dan masking sementara agar dapat digunakan
kembali ke fungsi lainnya
self.result_image = finalthresh.copy()
self.result_image_mask = res.copy()

```

```

        # Membuat fungsi threshold
        save_path = self.img_name.replace(".png",
            "_hasil_threshold.png").replace(".jpg", "_hasil_threshold.jpg").replace(".bmp",
            "_hasil_threshold.bmp") # mengatur agar file hasil tidak menimpa file asal
        cv2.imwrite(save_path, finalthresh)
        pixmap_source = QPixmap(save_path) #./Citra Hasil/Threshold.jpg
        self.hasil.setPixmap(pixmap_source)
        self.hasil.setScaledContents(True) # Mengatur agar gambar ditampilkan
        sesuai dengan ukuran QLabel
        if self.mask_check.isChecked(): #kondisi ketika checkbox masking
        dicentang
            mask_path = self.img_name.replace(".png",
            "_hasil_masking.png").replace(".jpg", "_hasil_masking.jpg").replace(".bmp",
            "_hasil_masking.bmp") # mengatur agar file hasil tidak menimpa file asal
            cv2.imwrite(mask_path, res)
            pixmap_mask = QPixmap(mask_path)
            self.hasil.setPixmap(pixmap_mask)
            self.hasil.setScaledContents(True) # Mengatur agar gambar ditampilkan
            sesuai dengan ukuran QLabel
        else: #kondisi ketika checkbox masking tidak dicentang
            self.hasil.setPixmap(pixmap_source)
            self.hasil.setScaledContents(True) # Mengatur agar gambar ditampilkan
            sesuai dengan ukuran QLabel

    def hitungRGB(self, totalColor):
        # Mengitung nilai RGB
        B = ((totalColor)[0])
        G = ((totalColor)[1])
        R = ((totalColor)[2])

        # Mengkonversi nilai RGB menjadi HSI
        self.r = R / (R + B + G)
        self.g = G / (R + B + G)
        self.b = B / (R + B + G)

        return self.r,self.g,self.b

    def hitungHSV(self,r,g,b):
        self.H = 0
        self.S = 0
        self.I = 0

        Max = max(r,g,b)
        Min = min(r,g,b)
        Delta = Max - Min

```

```

if Max == 0:
    self.S = 0
else:
    self.S = (Delta / Max) * 100
self.I = Max * 100
if Max == Min:
    self.H = 0
elif Max == r:
    self.H = (60 * ((g - b) / Delta) + 360) % 360
elif Max == g:
    self.H = (60 * ((b - r) / Delta) + 120) % 360
elif Max == b:
    self.H = (60 * ((r - g) / Delta) + 240) % 360

return self.H,self.S,self.I

def hitungDimensiCitra(self,g):
    # Menghitung nilai pixel
    P=int(self.count)

    # img=cv2.imread("tes hasil.png")
    # # Load Object Detector
    # detector = HomogeneousBgDetector()
    # ...
    # contours = detector.detect_objects(img)

    # # Draw objects boundaries
    # for cnt in contours:
    #     # Get rect
    #     rect = cv2.minAreaRect(cnt)
    #     (x, y), (w, h), angle = rect
    #     diameter_mayor = (w*3)/((8088)**(1/2))
    #     diameter_minor = (h*3)/((8088)**(1/2))
    #     # Display rectangle
    #     box = cv2.boxPoints(rect)
    #     box = np.int0(box)

    # Hitung tinggi dan lebar buah dalam cm
    tinggi_cm = (0.00006*(P)) + 3.3663
    lebar_cm = (0.00005*(P)) + 3.2945
    # tinggi_cm = diameter_mayor
    # lebar_cm = diameter_minor
    # Menghitung luas dan massa
    area = (0.0011*P) + 4.8091
    massa = (0.0035*P) - 26.802
    #format angka desimal

```



```

tinggi_cm =(f'{tinggi_cm:.2f} cm')
lebar_cm =(f'{lebar_cm:.2f} cm')
area =(f'{area:.2f} cm\u00b2')
massa =(f'{massa:.2f} gram')

# Segmentasi warna jeruk pada citra
#matang
# lower_red = np.array([0, 100, 100])
# upper_red = np.array([30, 255, 255])
lower_orange = np.array([11, 70, 41])
upper_orange = np.array([105, 255, 255])
mask1 = cv2.inRange(self.hsv, lower_orange, upper_orange)
#mentah
# lower_red = np.array([160, 100, 100])
# upper_red = np.array([179, 255, 255])
lower_green = np.array([0, 46, 17])
upper_green = np.array([109, 255, 255])
mask2 = cv2.inRange(self.hsv, lower_green, upper_green)

mask = mask1 + mask2

# Dilasi mask dengan kernel berukuran 3x3
kernel = np.ones((3,3),np.uint8)
dilation = cv2.dilate(mask,kernel,iterations = 1)
# Erosi mask dengan kernel berukuran 3x3
erosion = cv2.erode(mask,kernel,iterations = 1)
# Smoothing citra dengan Gaussian Blur
blur = cv2.GaussianBlur(self.img,(3,3),0)

# Hitung nilai kemanisan pada citra
B_val = self.b
V = 20.03*B_val + 5.8435
V =(f'{V:.2f} % brix')

# Menghitung kekerasan
B_val = self.b
keras = 6.6847 * B_val + 0.8555
keras = (f'{keras:.2f} kgf')

return tinggi_cm,lebar_cm,area,massa,V,keras

"""def Save(self):
    cv2.imshow('image',output)
    cv2.imwrite('./Citra_Hasil.jpg', output)
    pixmap_source = QPixmap('./Citra_Hasil.jpg')
    self.hasil.setPixmap(pixmap_source)"""

```

```

# Membuat perintah untuk menyimpan gambar
def Save(self):
    save_path, _ = QFileDialog.getSaveFileName(self, 'Simpan File', "", 'Images
(*.png *.jpg *.jpeg *.bmp)')
    if save_path:
        if self.mask_check.isChecked(): #mengatur agar hasil masking dapat
disimpan apabila checkbox mask dicentang
            if self.result_image is not None:
                cv2.imwrite(save_path, self.result_image_mask)
            else: #mengatur agar hanya hasil threshold yang dapat disimpan apabila
checkbox mask tidak dicentang
                if self.result_image is not None:
                    #self.saveToCSV(self.PictureName,self.GroupName,self.r,self.g,self.
b,self.H,self.S,self.I,self.count)
                    cv2.imwrite(save_path, self.result_image)

                #self.statusBar().showMessage(f"Gambar berhasil disimpan sebagai
{save_path}")
            #else:
                #self.statusBar().showMessage("Error: Gambar tidak terbaca atau
kosong")

def SaveToExcel(self):
    if self.row_counter == 1:
        # Menambahkan header jika baris pertama kosong
        headers = ["Nama File", "Red", "Green", "Blue", "Pixel Count", "Hue",
"Saturation", "Intensity", "Tinggi (cm)", "Lebar (cm)", "Luas (cm^2)", "Massa
(gram)", "Brix (%)", "Kekerasan (kgf)"]
        self.sheet.append(headers)
        self.row_counter += 1 #beralih ke baris berikutnya setelah headers terisi

    data_row = [
        self.PictureName,
        f"{float(self.red_val.text()):.5f}", # 15 angka di belakang koma untuk
mempertahankan jumlah nilai desimal asli
        f"{float(self.green_val.text()):.5f}",
        f"{float(self.blue_val.text()):.5f}",
        self.pixel_val.text(),
        self.hue_val.text(),
        self.sat_val.text(),
        self.intens_val.text(),
        self.mayor_val.text(),
        self.minor_val.text(),
        self.luas_val.text(),
        self.massa_val.text(),
        self.brix_val.text(),

```

```

        self.kekerasan_val.text(),
    ]
    self.sheet.append(data_row)
    # menyimpan hasil analisis citra ke dalam file excel yang sudah tersedia
    self.workbook.save("./datamentahRobotik.xlsx")

def Keluar(self):
    sys.exit(app.exec_())

# def convert_to_255(value):
#     if np.isnan(value):
#         # Nilai default jika value adalah NaN
#         return 0 # Misalnya, mengembalikan nilai 0
#     else:
#         return int(value * 255)

app=QApplication(sys.argv)
mainwindow=Robotik()
widget=QtWidgets.QStackedWidget()
widget.addWidget(mainwindow)
# widget.setFixedWidth(1920)
# widget.setFixedHeight(980)
widget.setWindowTitle('Robotik Nibroos')b
widget.setGeometry(0,0,1920,980)
widget.show()
sys.exit(app.exec_())

```