

**IMPLEMENTASI *ADAPTIVE NEURO-FUZZY INFERENCE*
SYSTEM (ANFIS) UNTUK DETEKSI OTOMATIS JENIS BUAH
BERDASARKAN CITRA WARNA DAN BENTUK
MENGUNAKAN MATLAB**

TUGAS AKHIR

Sebagai Salah Satu Syarat Untuk Memperoleh Gelar S1

Pada Program Studi Teknik Elektro

Universitas Islam Sultan Agung



DISUSUN OLEH :

Ihza Reza Fachrezzy S.A

NIM. 30602000007

PROGRAM STUDI TEKNIK ELEKTRO

FAKULTAS TEKNOLOGI INDUSTRI

UNIVERSITAS ISLAM SULTAN AGUNG

SEMARANG

2025

***IMPLEMENTATION OF ADAPTIVE NEURO-FUZZY
INFERENCE SYSTEM (ANFIS) FOR AUTOMATIC FRUIT TYPE
DETECTION BASED ON COLOR AND SHAPE IMAGES USING
MATLAB***

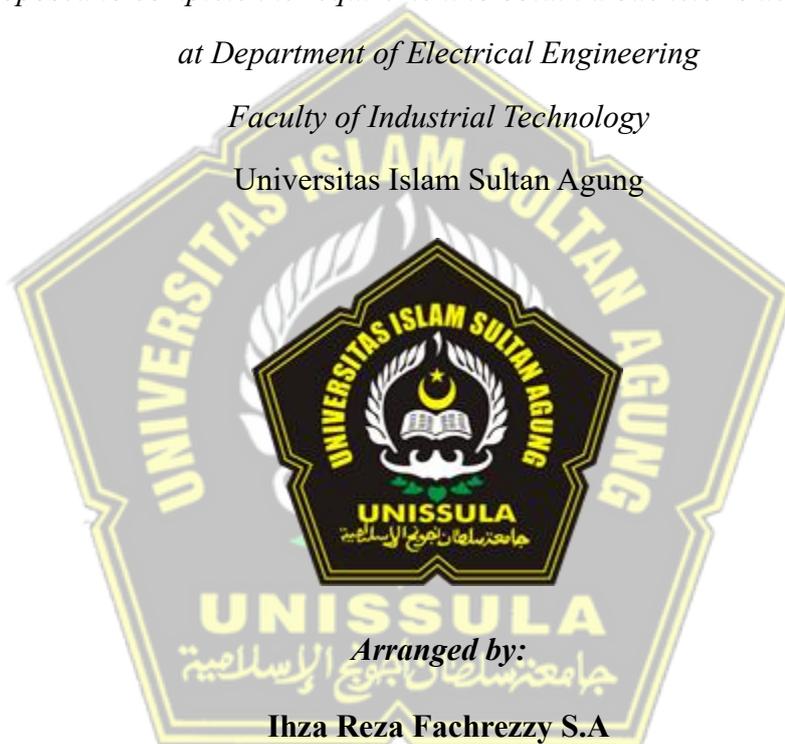
FINAL PROJECT

Proposed to complete the requirement to obtain a bachelor's degree (SI)

at Department of Electrical Engineering

Faculty of Industrial Technology

Universitas Islam Sultan Agung



Arranged by:

Ihza Reza Fachrezzy S.A

NIM. 30602000007

DEPARTMENT OF ELECTRICAL ENGINEERING

FACULTY OF INDUSTRIAN TECHNOLOGY

UNIVERSITAS ISLAM SULTAN AGUNG

SEMARANG

2025

LEMBAR PENGESAHAN PEMBIMBING

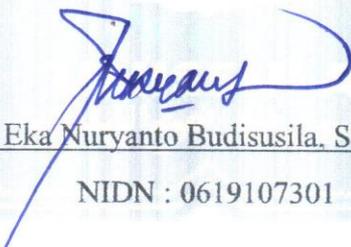
Laporan Tugas Akhir dengan judul “**IMPLEMENTASI *ADAPTIVE NEURO-FUZZY INFERENCE SYSTEM* (ANFIS) UNTUK DETEKSI OTOMATIS JENIS BUAH BERDASARKAN CITRA WARNA DAN BENTUK MENGGUNAKAN MATLAB**” ini disusun oleh:

Nama : IHZA REZA FACHREZZY S.A
NIM : 30602000007
Program Studi : Teknik Elektro

Telah disahkan dan disetujui oleh dosen pembimbing pada:

Hari : **Senin**
Tanggal : **3 Maret 2025**

Pembimbing


Dr. Eka Nuryanto Budisusila, S.T., M.T.

NIDN : 0619107301

Mengetahui,

Ka. Program Studi Teknik Elektro


Jenny Putri Hapsari, S.T., M.T.

NIDN : 0607018501

LEMBAR PENGESAHAN PENGUJI

Laporan Tugas Akhir dengan judul "**IMPLEMENTASI *ADAPTIVE NEURO-FUZZY INFERENCE SYSTEM* (ANFIS) UNTUK DETEKSI OTOMATIS JENIS BUAH BERDASARKAN CITRA WARNA DAN BENTUK MENGGUNAKAN MATLAB**" ini telah dipertahankan di depan Penguji sidang Tugas Akhir pada:

Hari : **Senin**
Tanggal : **3 Maret 2025**

Tim Penguji

Tanda Tangan

Agus Suprajitno, S.T., M.T.

NIDN : 0602047301

Ketua Penguji



.....

Dr. Bustanul Arifin, S.T., M.T.

NIDN : 0614117701

Penguji I

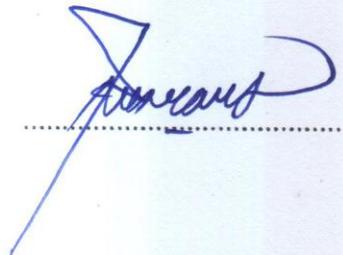


.....

Dr. Eka Nuryanto Budisusila, S.T., M.T.

NIDN : 0619107301

Penguji II



.....

SURAT PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan dibawah ini:

Nama : IHZA REZA FACHREZZY S.A

NIM : 30602000007

Jurusan : Teknik Elektro

Fakultas : Fakultas Teknologi Industri

Dengan ini saya menyatakan bahwa Tugas Akhir yang diajukan dengan judul **“IMPLEMENTASI *ADAPTIVE NEURO-FUZZY INFERENCE SYSTEM* (ANFIS) UNTUK DETEKSI OTOMATIS JENIS BUAH BERDASARKAN CITRA WARNA DAN BENTUK MENGGUNAKAN MATLAB”** adalah hasil karya sendiri, tidak pernah diajukan untuk memperoleh gelar kesarjanaan di perguruan tinggi lain maupun ditulis dan diterbitkan orang lain, kecuali secara tertulis diacu dalam daftar pustaka. Tugas Akhir ini adalah milik saya segala bentuk kesalahan dan kekeliruan dalam Tugas Akhir ini adalah tanggung jawab saya.

Demikian surat pernyataan ini saya buat dengan sadar dan penuh tanggung jawab.

Semarang, 16 Januari 2025

Yang Menyatakan



Ihza Reza Fachrezzy S.A

PERNYATAAN PERSETUJUAN PUBLIKASI ILMIAH

Yang bertanda tangan di bawah ini :

Nama : IHZA REZA FACHREZZY S.A
NIM : 30602000007
Jurusan : Teknik Elektro
Fakultas : Fakultas Teknologi Industri

Dengan ini saya menyerahkan karya ilmiah berupa Tugas Akhir dengan judul **“IMPLEMENTASI *ADAPTIVE NEURO-FUZZY INFERENCE SYSTEM* (ANFIS) UNTUK DETEKSI OTOMATIS JENIS BUAH BERDASARKAN CITRA WARNA DAN BENTUK MENGGUNAKAN MATLAB”** dan menyetujui menjadi hak milik Universitas Islam Sultan Agung serta memberikan hak bebas royalti non-eksklusif untuk disimpan, dialih mediakan, dikelola dalam pangkalan data dan publikasinya di internet atau media lain untuk kepentingan akademis selama tetap mencantumkan nama penulis sebagai pemilik Hak Cipta.

Pernyataan ini saya buat dengan sungguh-sungguh. Apabila dikemudian hari terbukti ada pelanggaran Hak Cipta/ Plagiarisme dalam karya ilmiah ini, maka segala bentuk tuntutan hukum yang timbul akan saya tanggung secara pribadi tanpa melibatkan pihak Universitas Islam Sultan Agung

Semarang, 16 Januari 2025

Yang Menyatakan

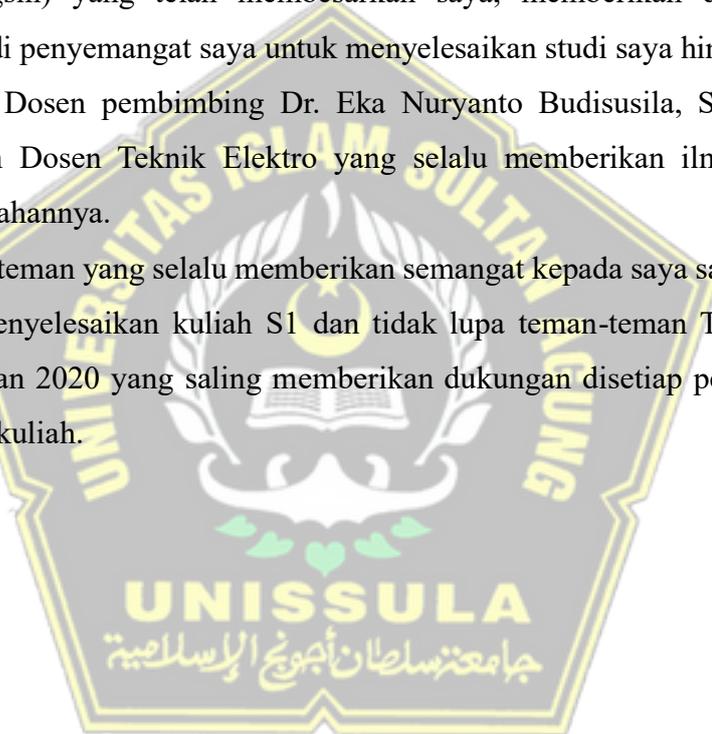


Ihza Reza Fachrezzy S.A

HALAMAN PERSEMBAHAN

Dengan rasa syukur yang mendalam, sungguh perjuangan yang cukup panjang telah aku lalui untuk mendapat gelar sarjana ini. Rasa syukur dan bahagia yang kurasakan ini akan ku persembahkan kepada :

1. Tugas akhir ini akan saya persembahkan kepada kedua orang tua saya yang saya cintai dan banggakan (Bapak Muhamad Ari Yusuf & Ibu Retna Winingsih) yang telah membesarkan saya, memberikan dukungan dan menjadi penyemangat saya untuk menyelesaikan studi saya hingga saat ini.
2. Untuk Dosen pembimbing Dr. Eka Nuryanto Budisusila, S.T., M.T. dan seluruh Dosen Teknik Elektro yang selalu memberikan ilmu, saran dan pengarahannya.
3. Untuk teman yang selalu memberikan semangat kepada saya saat menempuh dan menyelesaikan kuliah S1 dan tidak lupa teman-teman Teknik Elektro angkatan 2020 yang saling memberikan dukungan disetiap perjalanan saya dalam kuliah.



HALAMAN MOTTO

Scientia potentia est.

“Pengetahuan adalah kekuatan.”

(Sir Francis Bacon)

“Teknologi diciptakan untuk membantu manusia, tetapi inovasi lahir dari kemauan untuk terus belajar dan berkembang.”



KATA PENGANTAR

Assalamu'alaikum Warahmatullahi Wabarakatuh

Segala puji bagi Allah yang telah memberikan Taufik, Rahmat dan Hidayat-Nya kepada kita, sehingga penulis dapat menyelesaikan Tugas Akhir ini dengan baik dan lancar. Penulisan Tugas Akhir ini adalah salah satu syarat yang menjadi kewajiban kita untuk meraih Gelar Sarjana (S1) Jurusan Teknik Elektro Fakultas Teknologi Industri Universitas Islam Sultan Agung Semarang, adapun judul Tugas Akhir ini adalah “Implementasi *Adaptive Neuro-Fuzzy Inference System* (ANFIS) Untuk Deteksi Otomatis Jenis Buah Berdasarkan Citra Warna dan Bentuk Menggunakan MATLAB”. Sholawat serta salam tak lupa kita panjatkan kepada junjungan Nabi Muhammad SAW yang telah mengantarkan kita kejalan yang lurus. Penulis menyadari bahwa selama proses penulisan terdapat banyak kendala, namun berkat bantuan, bimbingan, dan kerjasama dari berbagai pihak serta berkat Allah SWT sehingga kendala-kendala yang penulis hadapi tersebut dapat diatasi. Oleh karena itu, pada kesempatan ini penulis dengan penuh rasa tulus dan ikhlas mengucapkan banyak banyak terimakasih kepada :

1. Allah SWT yang telah memberikan rahmat dan ridhonya serta memberikan ketabahan, kesabaran dan kelapangan hati serta pikiran dalam menimba ilmu.
2. Bapak Prof. Dr. H. Gunarto, SH., MH selaku Rektor Universitas Islam Sultan Agung Semarang.
3. Ibu Dr. Ir. Novi Marlyana, ST., MT. selaku Dekan Fakultas Teknologi Industri, Universitas Islam Sultan Agung Semarang.
4. Ibu Jenny Putri Hapsari, S.T., MT. selaku Ketua Program Studi Teknik Elektro, Fakultas Teknologi Industri, Universitas Islam Sultan Agung Semarang.
5. Bapak Agus Suprajitno, S.T, M.T. selaku dosen wali Teknik Elektro 2020 yang telah memberikan arahan selama menempuh studi.

6. Terimakasih kepada Dosen Pembimbing saya Dr. Eka Nuryanto Budisusila, S.T, MT. yang telah memberikan bimbingan dan dukungan dalam penyusunan Tugas Akhir ini
7. Seluruh Dosen dan Karyawan Jurusan Teknik Elektro, Universitas Islam Sultan Agung Semarang atas ilmu, bimbingan, dan bantuannya sehingga penulis bisa menyelesaikan Tugas Akhir ini.
8. Kedua orang tua saya beserta segenap keluarga yang saya cintai yang telah memberikan dukungan baik materiil maupun non materiil dan tidak pernah berhenti mendo'akan saya.
9. Kepada sahabat seperjuangan saya, yaitu Mahasiswa Teknik Elektro angkatan 2020 yang membantu dalam pembuatan laporan Tugas Akhir ini.
10. Semua pihak yang tidak dapat penulis sebutkan satu per satu, atas dukungan, doa dan semangat kepada penulis dalam menyelesaikan Tugas Akhir ini.

Penulis menyadari bahwa Tugas Akhir ini masih jauh dari sempurna dikarenakan keterbatasan ilmu, pengalaman dan kemampuan penulis. Oleh karena itu, saran dan kritik yang membangun dari pembaca akan menjadi masukan yang sangat berharga bagi penulis. Semoga Tugas Akhir ini dapat bermanfaat bagi pembaca. Aamiin.

Wassalamu'alaikum Warahmatullahi Wabarakatuh

Semarang, 16 Januari 2025



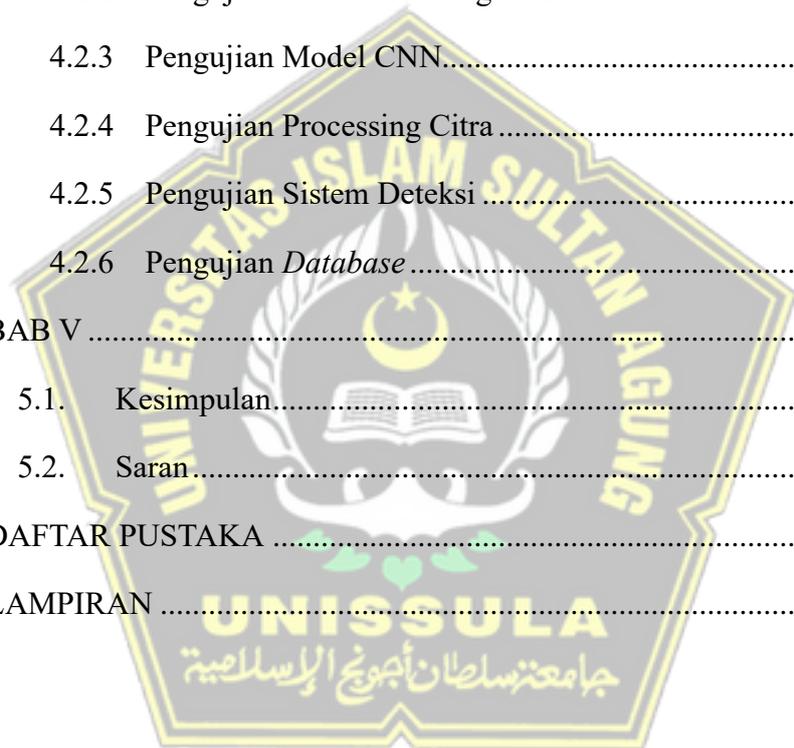
Ihza Reza Fachrezzy S.A

DAFTAR ISI

LEMBAR PENGESAHAN PEMBIMBING	ii
LEMBAR PENGESAHAN PENGUJI.....	iii
SURAT PERNYATAAN KEASLIAN TUGAS AKHIR	iv
PERNYATAAN PERSETUJUAN PUBLIKASI ILMIAH	v
HALAMAN PERSEMBAHAN	vi
HALAMAN MOTTO.....	vii
KATA PENGANTAR	viii
DAFTAR ISI.....	x
DAFTAR GAMBAR.....	xiii
DAFTAR TABEL	xv
ABSTRAK.....	xvi
<i>ABSTRACT</i>	xvii
BAB I.....	1
1.1. Latar Belakang Masalah.....	1
1.2. Perumusan Masalah.....	2
1.3. Pembatasan Masalah	3
1.4. Tujuan.....	4
1.5. Manfaat.....	4
1.6. Sistematika Laporan	5
BAB II	7
2.1. Tinjauan Pustaka.....	7
2.2. Landasan Teori	8
2.2.1 Pengolahan Citra Digital.....	8

2.2.2	<i>Adaptive Neuro-Fuzzy Inference System (ANFIS)</i>	10
2.2.3	Jaringan Syaraf Tiruan (<i>Neural Networks</i>).....	10
2.2.4	<i>Convolutional Neural Network (CNN)</i>	10
2.2.5	<i>Fuzzy Logic</i>	14
2.2.6	Deteksi Objek Berdasarkan Warna dan Bentuk.....	15
2.2.7	MATLAB Sebagai Alat Pengembangan	18
2.2.8	Python sebagai Penghubung	20
BAB III		22
3.1.	Metode Penelitian.....	22
3.2.	Analisis Kebutuhan Sistem.....	23
3.2.1	Kebutuhan Fungsional Sistem	23
3.2.2	Kebutuhan Non-Fungsional Sistem.....	24
3.3.	Rancangan Perangkat Keras	24
3.4.	Rancangan Perangkat Lunak	27
3.4.1	Platform	27
3.4.2	Modul Sistem.....	28
3.4.3	<i>Database</i>	44
3.5.	Rancangan Pengujian	45
3.5.1	Pengujian Perangkat Keras	46
3.5.2	Pengujian Perangkat Lunak	47
BAB IV		52
4.1.	Hasil Perancangan	52
4.1.1	Hasil Perancangan Perangkat Keras	52
4.1.2	Hasil Perancangan Akuisisi Gambar.....	53
4.1.3	Hasil Perancangan Processing Citra	54

4.1.4	Hasil Perancangan Training Data.....	56
4.1.5	Hasil Perancangan Proses Deteksi.....	62
4.1.6	Hasil Perancangan Interface	63
4.1.7	Hasil Perancangan <i>Database</i>	64
4.2.	Pengujian dan Analisis Hasil Pengujian.....	66
4.2.1	Pengujian Perangkat Keras	67
4.2.2	Pengujian Platform Perangkat Lunak	67
4.2.3	Pengujian Model CNN.....	68
4.2.4	Pengujian Processing Citra	73
4.2.5	Pengujian Sistem Deteksi	74
4.2.6	Pengujian <i>Database</i>	78
BAB V	81
5.1.	Kesimpulan.....	81
5.2.	Saran.....	82
DAFTAR PUSTAKA	84
LAMPIRAN	87



DAFTAR GAMBAR

Gambar 2. 1. Citra RGB.....	8
Gambar 2. 2. Citra <i>grayscale</i>	9
Gambar 2. 3. Citra biner.....	9
Gambar 2. 4. Struktur CNN	11
Gambar 2. 5. Jenis-jenis metode edge detection	16
Gambar 3. 1 Flowchart metode penelitian	22
Gambar 3. 2 Diagram blok perancangan hardware.....	24
Gambar 3. 3 Rancangan kamera dan objek.....	26
Gambar 3. 4 Rancangan studio <i>mini</i>	26
Gambar 3. 5. Flowchart keseluruhan sistem	28
Gambar 3. 6 Diagram blok modul akuisisi gambar	29
Gambar 3. 7 Tampilan IP address pada Droidcam Classic	29
Gambar 3. 8 Tampilan <i>input</i> IP address pada Droidcam Windows Client.....	30
Gambar 3. 9 Diagram blok <i>preprocessing</i>	30
Gambar 3. 10 Flowchart <i>postprocessing</i>	31
Gambar 3. 11 Flowchart <i>training</i> warna	33
Gambar 3. 12 Rancangan arsitektur CNN.....	35
Gambar 3. 13 Flowchart <i>training</i> bentuk.....	37
Gambar 3. 14 Rancangan <i>fuzzy</i>	39
Gambar 3. 15. Rancangan pembagian kelas <i>input fuzzy</i>	39
Gambar 3. 16. Rancangan <i>rules fuzzy</i>	40
Gambar 3. 17 Flowchart modul deteksi	41
Gambar 3. 18 Rancangan <i>output fuzzy</i>	42
Gambar 3. 19. Rancangan <i>Apps Designer</i>	43
Gambar 3. 20 Rancangan penyusunan folder <i>database</i>	45
Gambar 3. 21 Contoh isi dalam subfolder nama buah Pisang	45
Gambar 4. 1. Perangkat sistem deteksi otomatis jenis buah	52
Gambar 4. 2. Studio <i>mini</i> dengan pencahayaan tidak dinyalakan.....	53
Gambar 4. 3. Studio <i>mini</i> dengan pencahayaan dinyalakan.....	53

Gambar 4. 4. Live video pada Droidcam Windows Client	54
Gambar 4. 5. Live video pada Python.....	54
Gambar 4. 6. Citra asli dari kamera smartphone.....	55
Gambar 4. 7. Hasil <i>resize preprocessing</i>	55
Gambar 4. 8. Hasil citra akhir <i>preprocessing</i>	56
Gambar 4. 9. Hasil <i>postprocessing</i> citra	56
Gambar 4. 10. Grafik <i>training progress</i> CNN untuk deteksi warna	57
Gambar 4. 11. Hasil perhitungan nilai statistik citra.....	57
Gambar 4. 12. Box plot dan grafik hasil perhitungan nilai statistik citra	57
Gambar 4. 13. Kategori <i>input output fuzzy</i>	58
Gambar 4. 14. Diagram blok <i>rules fuzzy</i>	62
Gambar 4. 15. Tampilan hasil proses deteksi.....	63
Gambar 4. 16. Interface deteksi buah (live kamera dan <i>Apps Designer</i>).....	64
Gambar 4. 17. Hasil pengkategorian subfolder <i>database</i>	64
Gambar 4. 18. Gambar pada subfolder Alpukat.....	65
Gambar 4. 19. Gambar pada subfolder Apel.....	65
Gambar 4. 20. Gambar pada subfolder Jeruk.....	65
Gambar 4. 21. Gambar pada subfolder Pisang.....	66
Gambar 4. 22. Gambar pada subfolder Kiwi	66

UNISSULA
جامعنا سلطان أبجوج الإسلامية

DAFTAR TABEL

Tabel 3.1. Rancangan pengujian perangkat keras	46
Tabel 3. 2. Rancangan pengujian perangkat lunak.....	47
Tabel 4. 1. Hasil pengujian perangkat keras.....	67
Tabel 4. 2. Hasil pengujian platform perangkat lunak	68
Tabel 4. 3. Hasil pengujian dengan perubahan <i>input size</i>	69
Tabel 4. 4. Hasil pengujian dengan perubahan parameter pelatihan.....	70
Tabel 4. 4. Hasil pengujian dengan perubahan parameter pelatihan.....	71
Tabel 4. 5. Hasil pengujian berdasarkan latar belakang dan pencahayaan	73
Tabel 4. 6. Hasil pengujian sistem deteksi pada alpukat.....	74
Tabel 4. 6. Hasil pengujian sistem deteksi pada alpukat.....	75
Tabel 4. 7. Hasil pengujian sistem deteksi pada apel.....	75
Tabel 4. 8. Hasil pengujian sistem deteksi pada jeruk	76
Tabel 4. 9. Hasil pengujian sistem deteksi pada kiwi	77
Tabel 4. 10. Hasil pengujian sistem deteksi pada pisang	77
Tabel 4. 10. Hasil pengujian sistem deteksi pada pisang	78
Tabel 4. 11. Hasil pengujian dengan perubahan jumlah <i>database</i>	78
Tabel 4. 11. Hasil pengujian dengan perubahan jumlah <i>database</i>	79

ABSTRAK

Penelitian ini bertujuan untuk mengembangkan sistem deteksi otomatis jenis buah berdasarkan warna dan bentuk menggunakan *Adaptive Neuro-Fuzzy Inference System* (ANFIS) yang diimplementasikan pada platform MATLAB. Sistem ini memanfaatkan fitur dominan seperti warna serta bentuk (eksentrisitas dan area) untuk mendeteksi lima jenis buah, yaitu pisang, alpukat, apel, jeruk, dan kiwi. Data citra buah dikumpulkan menggunakan kamera *smartphone* dalam lingkungan terkendali, kemudian diproses menggunakan *Convolutional Neural Network* (CNN) untuk ekstraksi dan klasifikasi fitur warna. Sementara itu, fitur bentuk dievaluasi menggunakan metode *edge detection* dan analisis statistik.

Penelitian ini menggunakan pendekatan eksperimental, di mana sistem dirancang, diimplementasikan, dan diuji untuk mengukur akurasi deteksi. Model deteksi warna dan bentuk kemudian diintegrasikan menggunakan logika *fuzzy* untuk menghasilkan keputusan akhir. Hasil pengujian menunjukkan bahwa sistem ini mampu mendeteksi jenis buah dengan tingkat akurasi 99,8%, yang mengindikasikan potensinya dalam otomatisasi berbagai sektor, termasuk industri pangan, manufaktur, dan teknologi pengolahan citra. Penelitian ini memberikan kontribusi signifikan dalam pengembangan kecerdasan buatan untuk sistem pengenalan objek, sekaligus membuka peluang inovasi lebih lanjut dalam berbagai bidang teknologi..

Kata Kunci: ANFIS, *Neural Network*, *Convolutional Neural Network* (CNN), MATLAB, *Fuzzy Logic*.

ABSTRACT

This study aims to develop an automatic fruit classification system based on color and shape using the Adaptive Neuro-Fuzzy Inference System (ANFIS), implemented on the MATLAB platform. The system utilizes dominant features such as color and shape (eccentricity and area) to classify five types of fruit: banana, avocado, apple, orange, and kiwi. Fruit image data is collected using a smartphone camera in a controlled environment and processed using a Convolutional Neural Network (CNN) for feature extraction and color classification. Meanwhile, shape features are evaluated using the edge detection method and statistical analysis.

This study employs an experimental approach, where the system is designed, implemented, and tested to measure detection accuracy. The color and shape detection models are then integrated using fuzzy logic to produce a final decision. The test results indicate that the system can classify fruit types with 99.8% accuracy, demonstrating its potential for automation in various sectors, including the food industry, manufacturing, and image processing technology. This research contributes significantly to the development of artificial intelligence for object recognition systems, opening further opportunities for innovation in various technological fields.

Keywords: ANFIS, Neural Network, Convolutional Neural Network (CNN), MATLAB, Fuzzy Logic.

BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Buah merupakan bagian dari suatu tumbuhan yang berasal dari bunga atau putik dari tumbuhan tersebut dan biasanya memiliki biji sedangkan sayur merupakan daun, tumbuhan polong, atau biji-bijian yang dapat dimasak. Buah dan sayuran mengandung berbagai macam vitamin, mineral, dan serat pangan yang berperan untuk membantu proses metabolisme dalam tubuh. Buah dan sayuran memiliki berbagai macam jenis yang memiliki bentuk dan warna yang berbeda dari setiap jenisnya [1]. Klasifikasi dengan mengandalkan pengamatan visual manusia yang bersifat obyektif tentu memiliki beberapa kelemahan yakni, adanya keterbatasan visual manusia, dan sifat merasa gampang lelah dan jenuh ketika melakukan kegiatan yang terus berulang, dan kondisi psikis pengamat juga dapat mempengaruhi tingkat keakuratan dalam klasifikasi secara manual. Hal tersebut akan mengakibatkan proses klasifikasi tidak konsisten dan akan membutuhkan banyak waktu [2].

Klasifikasi buah merupakan salah satu penelitian yang cukup kompleks. Hal ini dikarenakan suatu jenis buah, apel misalnya dapat saja berbeda dengan apel lain dengan jenis yang sama dalam hal ukuran, warna, tekstur, bentuk dan lain-lain, selain itu persamaan bentuk, ukuran dan warna antar jenis buah juga menambah kompleksitas dalam proses klasifikasi buah. Buah sebagai objek yang akan diklasifikasi, memiliki banyak ciri yang dapat digunakan sebagai fitur. Salah satunya adalah fitur warna. Fitur warna dapat dikatakan sebagai ciri dominan dari suatu jenis buah yang dapat membedakan satu jenis buah dengan buah lainnya. Penggunaan fitur warna dalam proses klasifikasi buah telah banyak diterapkan pada penelitian terdahulu [3]. Namun jika hanya menggunakan fitur warna maka akurasi yang didapat cukup kurang, sehingga dibutuhkanlah fitur-fitur lain seperti fitur bentuk. Dalam hal bentuk sebuah objek terdapat berbagai macam hal yang dapat diamati seperti eksentrisitas dan luas.

Oleh karena itu, dalam upaya meningkatkan efisiensi dan akurasi dalam klasifikasi buah, penerapan teknologi digital menjadi sangat penting, terutama dengan berkembangnya revolusi industri 4.0 di Indonesia.. Indonesia sudah mulai memasuki revolusi industri 4.0, dimana para pengusaha yang menerapkan teknologi digital yang akan mampu bertahan. Kemajuan suatu industri harus didorong dari kemajuan teknologi digital para pelaku usaha. Berkembangnya teknologi digital di suatu industri akan mempengaruhi penerapan ekonomi digital pada suatu negara [4].

Dalam upaya mendukung otomatisasi industri, pengembangan sistem deteksi buah berbasis kecerdasan buatan menjadi solusi yang potensial. Sistem ini dapat diterapkan dalam berbagai sektor industri, terutama dalam proses penyortiran dan pengemasan buah secara otomatis di industri pangan, manufaktur, serta perdagangan. Dengan adanya sistem ini, klasifikasi buah dapat dilakukan dengan lebih cepat, akurat, dan konsisten, sehingga mengurangi ketergantungan pada tenaga kerja manual serta meningkatkan efisiensi produksi. Selain itu, teknologi ini juga memiliki potensi untuk diterapkan dalam berbagai bidang lain, seperti inspeksi kualitas produk dan manajemen rantai pasok berbasis visi komputer.

Berdasarkan permasalahan di atas, maka diperlukan suatu sistem yang mampu membedakan jenis jenis buah. Dengan memanfaatkan kemajuan ilmu pengetahuan yang pesat khususnya di bidang kecerdasan buatan yang dikolaborasikan dengan pengolahan citra digital, mampu menghasilkan suatu sistem yang dapat mengklasifikasikan jenis jenis buah. Pengolahan citra digital merupakan disiplin ilmu yang mempelajari tentang bagaimana mengolah citra, pengolahan citra dapat dikembangkan dengan tujuan *image enhancement*, *image compression*, *image restoration*, dan *feature extraction* [2].

1.2. Perumusan Masalah

1. Bagaimana penerapan metode *Adaptive Neuro-Fuzzy Inference System* (ANFIS) dalam mengintegrasikan fitur warna dan bentuk untuk meningkatkan performa klasifikasi buah?

2. Seberapa efektif kombinasi *Convolutional Neural Network* (CNN) untuk ekstraksi fitur warna dan metode *Edge Detection* untuk analisis bentuk dalam meningkatkan akurasi sistem?
3. Sejauh mana tingkat akurasi sistem deteksi yang dihasilkan?
4. Apakah MATLAB dapat digunakan untuk simulasi sistem?

1.3. Pembatasan Masalah

Berdasarkan latar belakang yang telah dibahas sebelumnya, didapatkan batasan-batasan masalah penelitian ini , yaitu sebagai berikut:

1. Jenis buah yang dideteksi, penelitian ini hanya akan fokus pada 5 jenis buah, yaitu pisang, alpukat, apel, jeruk dan kiwi.
2. Fitur warna dan bentuk, deteksi hanya akan didasarkan pada fitur warna dan bentuk buah. Faktor lain seperti tekstur dan berat buah tidak akan dipertimbangkan.
3. Jarak pengambilan citra ditetapkan pada jarak 15cm sampai 20cm dari permukaan buah menyesuaikan frame pada tangkapan kamera.
4. Sudut pandang capture objek konsisten dari depan, sudut pandang dari sisi lain tidak dibahas.
5. Jumlah data latih, *database* yang digunakan untuk melatih dan menguji model jaringan syaraf tiruan dibatasi pada sejumlah 90 data tiap jenis buah.
6. Teknik jaringan syaraf tiruan, penelitian ini hanya akan menggunakan satu jenis arsitektur jaringan syaraf tiruan, yaitu *Convolutional Neural Network* (CNN). Arsitektur lain tidak akan dieksplorasi dalam penelitian ini.
7. Platform pemrograman, implementasi dan pengujian model dilakukan menggunakan perangkat lunak MATLAB dan Python sebagai pendukung. Perangkat lunak atau bahasa pemrograman lain tidak akan digunakan dalam penelitian ini.
8. Penggunaan model, model yang dikembangkan dalam penelitian ini hanya akan digunakan untuk tujuan demonstrasi dan evaluasi. Penerapan dalam skenario industri atau aplikasi komersial tidak termasuk dalam cakupan penelitian.

1.4. Tujuan

1. Mengembangkan sistem deteksi otomatis jenis buah berdasarkan warna dan bentuk menggunakan metode *Adaptive Neuro-Fuzzy Inference System* (ANFIS).
2. Mengintegrasikan metode *Convolutional Neural Network* (CNN) untuk ekstraksi fitur warna dan metode *Edge Detection* untuk analisis bentuk dalam proses klasifikasi buah.
3. Mengevaluasi tingkat akurasi sistem deteksi dan menguji efektivitas sistem dalam mengklasifikasikan jenis buah.
4. Mengimplementasikan model tersebut menggunakan MATLAB.

1.5. Manfaat

Penelitian ini memiliki manfaat baik dari segi akademik maupun praktis dalam penerapan di industri. Dari segi akademik, penelitian ini diharapkan dapat menambah wawasan dan pengembangan ilmu di bidang kecerdasan buatan, khususnya dalam penerapan *Adaptive Neuro-Fuzzy Inference System* (ANFIS) untuk klasifikasi objek berdasarkan warna dan bentuk. Selain itu, penelitian ini memberikan kontribusi dalam pengembangan metode kombinasi *Convolutional Neural Network* (CNN) untuk ekstraksi fitur warna dan metode *edge detection* untuk analisis bentuk. Hasil penelitian ini juga dapat menjadi referensi bagi penelitian selanjutnya yang ingin mengembangkan sistem klasifikasi berbasis pengolahan citra digital dan kecerdasan buatan.

Dari segi praktis, sistem yang dikembangkan dalam penelitian ini dapat mempermudah proses klasifikasi buah secara otomatis dengan tingkat akurasi yang tinggi, sehingga dapat mengurangi ketergantungan pada metode manual yang lebih lambat dan kurang konsisten. Selain itu, sistem ini dapat mendukung otomatisasi dalam industri penyortiran buah, baik di tingkat pertanian, distribusi, hingga manufaktur pengolahan produk berbasis buah. Dengan adanya sistem klasifikasi otomatis ini, potensi kesalahan manusia dalam proses penyortiran dapat diminimalkan, sehingga meningkatkan efisiensi produksi dan mempercepat proses klasifikasi di sektor industri pangan dan perdagangan. Lebih jauh lagi, teknologi ini

memiliki potensi untuk diterapkan dalam berbagai sektor industri lain yang membutuhkan sistem inspeksi berbasis visi komputer, seperti industri pertanian, manufaktur, dan logistik.

Dengan adanya penelitian ini, diharapkan teknologi klasifikasi buah berbasis kecerdasan buatan dapat lebih dioptimalkan dan dikembangkan untuk mendukung berbagai kebutuhan industri serta menjadi dasar bagi inovasi teknologi di masa depan.

1.6. Sistematika Laporan

Dalam penyusunan tugas akhir ini dilakukan pengelompokan menurut isi dalam beberapa bab. Sistematika penulisannya adalah sebagai berikut.

BAB I PENDAHULUAN

Pada bab awal penelitian ini terdapat uraian mengenai latar belakang masalah, perumusan masalah, pembatasan masalah, manfaat penelitian dan tujuan penelitian yang dikerjakan dan sistematika penulisan laporan tugas akhir. Bab ini penting untuk menjelaskan bab-bab selanjutnya.

BAB II TINJAUAN PUSTAKA

Bab kedua ini berisi tentang tinjauan pustaka dan teori yang mendukung sistem ANFIS untuk mendeteksi jenis buah berdasarkan warna dan bentuk menggunakan MATLAB khususnya metode CNN.

BAB III METODE PENELITIAN

Bab ketiga ini membahas tahap perancangan sistem ANFIS pendeteksi jenis buah yang berisi uraian tahapan penyelesaian tugas akhir yang memuat tentang tahap perancangan sistem menggunakan platform MATLAB dan dengan bantuan Python dengan platform VSCODE, perancangan sistem untuk menghubungkan kamera dengan MATLAB. Tahap pembuatan desain CNN untuk melatih dan menguji data, pembuatan desain *interface* pada MATLAB untuk memudahkan pengguna dalam menjalankan sistem, serta perancangan desain *hardware* untuk memudahkan sistem mendeteksi citra buah. Pengintegrasian masing-masing fungsi

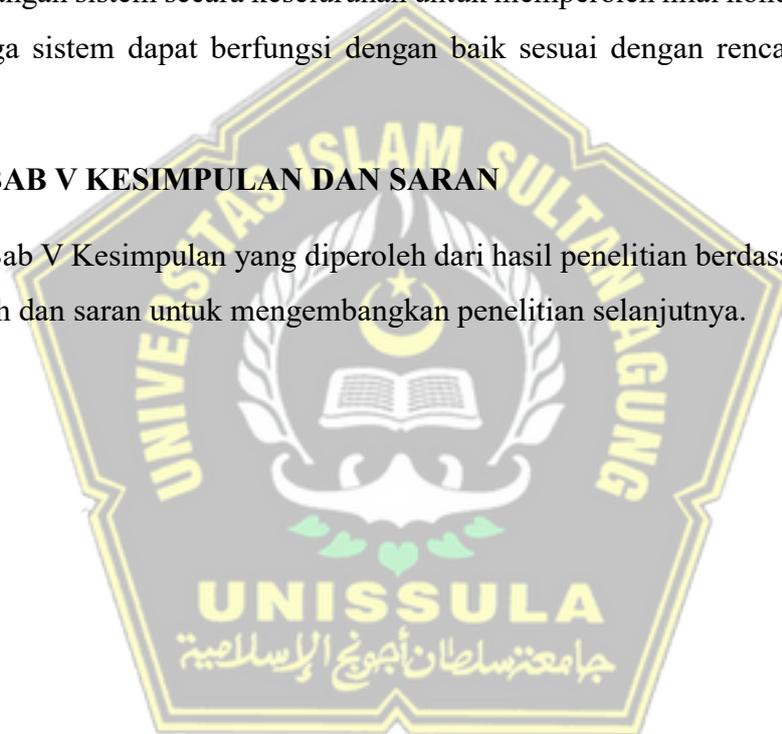
kontrol. Kemudian dilakukan perancangan perangkat keras yang telah didesain. Dan dilakukan tahap uji coba sistem yang telah terintegrasi.

BAB IV HASIL PERANCANGAN DAN PENGUJIAN

Bab keempat ini membahas langkah-langkah dalam pengujian sistem, hasil keakuratan dalam setiap uji coba, pengujian dengan sampel jenis buah yang berbeda, spesifikasi alat yang digunakan, serta petunjuk-petunjuk pengoperasian pada sistem yang dibuat. Selanjutnya, akan dibahas hasil pengujian dari perancangan sistem secara keseluruhan untuk memperoleh nilai kondisi yang tepat, sehingga sistem dapat berfungsi dengan baik sesuai dengan rencana yang telah dibuat.

BAB V KESIMPULAN DAN SARAN

Bab V Kesimpulan yang diperoleh dari hasil penelitian berdasarkan rumusan masalah dan saran untuk mengembangkan penelitian selanjutnya.



BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1. Tinjauan Pustaka

Tinjauan pustaka ini tidak lepas dari referensi dan digunakan sebagai pembandingan dari penelitian sebelumnya, berikut beberapa penelitian sebelumnya yang memiliki sistem yang sama.

Pada penelitian "*Dimensionality Reduction and Classification of Color Features Data Using SVM and kNN*"[5] klasifikasi buah cabai dilakukan dengan menggunakan kombinasi SVM dan kNN (*k-Nearest Neighbor*). Fitur yang digunakan adalah nilai rata-rata dari *channel Red* dan *channel Green* dari *Red Green Blue* (RGB). Penelitian ini berhasil mencapai tingkat akurasi sebesar 97.59% untuk SVM-kNN dengan *City Block* dan 97,36% untuk SVM-kNN dengan *Euclidean Distance*. *Artificial Neural Network* digunakan pada penelitian "*Identification and Classification of Bulk Fruits Images Using Artificial Neural Networks*"[6] dalam melakukan klasifikasi tumpukan buah (*bulk fruit*). Fitur yang digunakan adalah fitur model warna HSI dan *shape* dengan tingkat akurasi antara 92% hingga 94%. Klasifikasi tingkat kualitas buah belimbing dilakukan dengan menggunakan fitur model warna YCbCr sebagai fitur pada penelitian "*Starfruit Color Maturity Classification Using Cr as Feature*"[7]. Metode klasifikasi yang digunakan yaitu *Maturity Classification based on YCbCr Color Space* dan menghasilkan tingkat akurasi sebesar 96%. Penelitian untuk klasifikasi kualitas buah dengan jenis apel juga dilakukan pada penelitian "*Apples Grading Based on SVM Classifier*" [8]. Pada "*Apples Grading Based on SVM Classifier*" [8] metode klasifikasi yang digunakan adalah SVM dengan menggunakan fitur rata-rata nilai *channel Red* dan *Channel Green* dari model warna RGB. Tingkat akurasi yang dihasilkan yaitu sebesar 100%. Pada "*Classification of Fruits Using Probabilistic Neural Networks-Improvement Using Color Features*" [9] dilakukan penelitian klasifikasi buah dengan menggunakan metode *Probabilistic Neural Network* dengan tingkat keberhasilan antara 79-90%. Fitur yang digunakan adalah 17 fitur yaitu fitur

morfologi terdiri dari 3 fitur *size* dan 2 fitur *shape* dan fitur model warna RGB (6 fitur) dan HSI (6 fitur). kNN digunakan pada penelitian “*A New Method for Fruits Recognition System*” [10] untuk melakukan klasifikasi buah. Fitur yang digunakan adalah fitur model warna RGB dan fitur *shape*. Fitur model warna RGB yang digunakan adalah nilai minimum dan maksimum dari *Channel Red* dan *Green*, sedangkan fitur *shape* menggunakan 3 fitur, terdiri dari *roundness*, *area* dan *perimeter*. Tingkat akurasi yang dihasilkan adalah mencapai angka 90%.

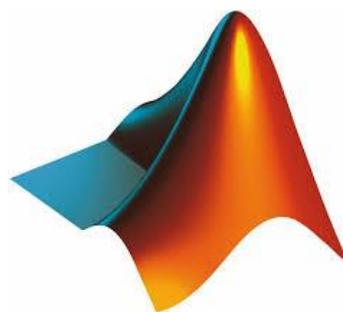
2.2. Landasan Teori

2.2.1 Pengolahan Citra Digital

Citra atau lebih dikenal dengan gambar dapat diartikan sebagai representasi atau kemiripan suatu objek. Citra dibagi menjadi dua macam: citra analog dan citra digital [11]. Teknologi pengolahan citra digunakan untuk mempersiapkan data sebelum diolah oleh model *neural network*. Studi meliputi metode *cropping*, *resizing*, konversi citra ke skala warna tertentu (RGB, *grayscale*), hingga deteksi tepi menggunakan metode Roberts. Citra digital dibagi menjadi 3 jenis berdasarkan intensitasnya, antara lain:

a. Citra Warna (RGB)

Salah satu jenis citra berwarna adalah citra 8 bit, dimana citra 8 bit ini memiliki kriteria setiap piksel dari citra warna diwakili oleh 8 bit, jumlah warna maksimum 256 warna. Citra warna memiliki matriks tiga dimensi, yaitu warna merah (*red*), hijau (*green*), dan biru (*blue*).



Gambar 2. 1. Citra RGB

b. Citra *Grayscale*

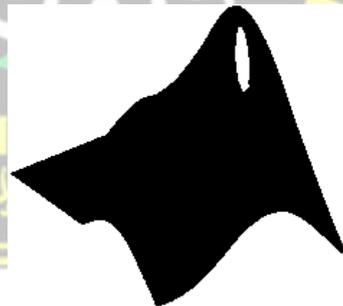
Citra *grayscale* adalah citra atau gambar yang memiliki matriks satu dimensi dimana matriks tersebut berisi nilai yang menyatakan intensitas warna keabuan.



Gambar 2. 2. Citra *grayscale*

c. Citra Biner

Citra biner atau citra B&W (*black and white*) adalah citra yang memiliki matriks satu dimensi dimana matriks tersebut hanya berisi nilai 1(putih) dan 0 (hitam) .



Gambar 2. 3. Citra biner

Pengolahan citra digital merujuk kepada cara memproses citra dua dimensi mempergunakan media computer [12]. Citra digital harus direpresentasikan (digambarkan) dalam bentuk angka numerik dengan nilai-nilai secara diskrit atau disebut dengan nilai intensitas cahaya, tujuannya agar komputer dapat mengolah citra digital. Nilai-nilai intensitas cahaya yang dimaksud digambarkan sebagai nilai-nilai kanal pada sebuah citra digital. Citra yang berukuran 8 bit mempunyai 1 kanal berisi sekelompok nilai dalam rentang nilai 0 – 255, pada citra 16 bit

mempunyai 2 kanal, dan pada citra yang berukuran 24 bit mempunyai 3 kanal yang disebut dengan istilah kanal RGB dimana R (*red*), G (*green*), dan B (*blue*).

2.2.2 Adaptive Neuro-Fuzzy Inference System (ANFIS)

Adaptive Neuro-Fuzzy Inference System (ANFIS) adalah metode kecerdasan buatan yang menggabungkan keunggulan jaringan saraf tiruan (*Artificial Neural Network/ANN*) dan sistem logika *fuzzy* untuk meningkatkan akurasi dalam pemodelan dan pengambilan keputusan. ANFIS menggunakan kemampuan pembelajaran dari jaringan saraf tiruan untuk mengoptimalkan parameter dalam sistem *fuzzy*, sehingga dapat secara adaptif menyesuaikan aturan *fuzzy* berdasarkan data pelatihan. Struktur ANFIS terdiri dari beberapa lapisan, termasuk fuzzifikasi, aturan *fuzzy*, inferensi, normalisasi, dan defuzzifikasi, yang bekerja secara terintegrasi untuk memproses *input* dan menghasilkan *output* yang lebih presisi. Dengan keunggulannya dalam menangani ketidakpastian dan kompleksitas data, ANFIS banyak digunakan dalam berbagai aplikasi, termasuk klasifikasi, prediksi, dan sistem kontrol.

2.2.3 Jaringan Syaraf Tiruan (Neural Networks)

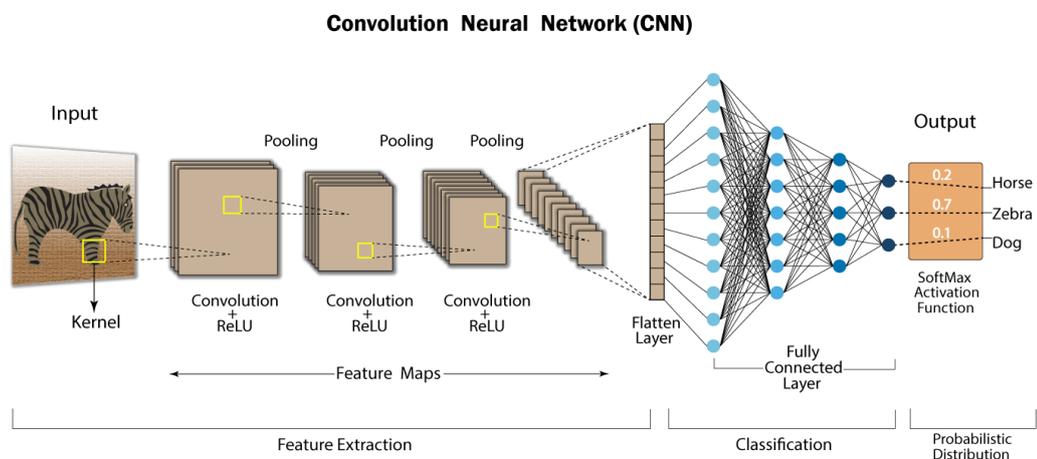
Jaringan syaraf tiruan adalah model komputasi yang terinspirasi oleh cara kerja otak manusia. Model ini terdiri dari sejumlah besar neuron buatan yang diatur dalam lapisan-lapisan dan saling terhubung. Setiap neuron menerima *input*, melakukan operasi sederhana, dan mengirim *output* ke neuron lain di lapisan berikutnya. Jaringan syaraf tiruan mampu belajar dari data dengan menyesuaikan bobot koneksi antar neuron melalui algoritma pembelajaran, seperti backpropagation [13].

2.2.4 Convolutional Neural Network (CNN)

Metode CNN (*Convolutional Neural Network*) merupakan sebuah pendekatan dalam dunia kecerdasan buatan (*artificial intelligence*) yang dirancang khusus untuk menangani data berstruktur seperti citra dan data spasial. CNN banyak digunakan dalam tugas-tugas pengolahan citra dan pengenalan pola, dan telah mencapai kesuksesan besar dalam berbagai aplikasi, termasuk klasifikasi

gambar, deteksi objek, dan segmentasi gambar. CNN termasuk dalam jenis *Deep Neural Network* karena kedalaman jaringan yang tinggi dan banyak diaplikasikan pada data citra [14]. *Convolutional Neural Network (CNN)* dapat digunakan untuk deteksi jenis buah dengan MATLAB. CNN adalah jenis jaringan syaraf tiruan yang sangat efektif untuk tugas-tugas pengenalan gambar dan klasifikasi objek, termasuk deteksi jenis buah berdasarkan gambar. Menurut [15] metode *Convolutional Neural Network* memiliki hasil yang paling signifikan dalam pengenalan citra digital. Hal tersebut dikarenakan CNN diimplementasikan berdasarkan sistem pengenalan citra pada visual cortex manusia.

CNN telah banyak digunakan dalam penelitian untuk mendeteksi dan mengklasifikasikan buah. [16] mengembangkan model CNN untuk klasifikasi berbagai jenis buah dengan menggunakan *database* gambar buah yang beragam. Penelitian ini menunjukkan bahwa CNN mampu mencapai akurasi tinggi dalam deteksi buah, mengatasi tantangan seperti variasi dalam pencahayaan dan latar belakang. Penelitian lain oleh [17] menunjukkan efektivitas CNN dalam mendeteksi buah dalam kondisi yang beragam, menggunakan teknik augmentasi data untuk meningkatkan kinerja model.



Gambar 2. 4. Struktur CNN

Pada CNN terdapat beberapa *layer* pada arsitekturnya, seperti *convolution layer*, *batch normalization layer*, *ReLU layer*, *softmax layer*, *max pooling* dan juga *fully connected layer*.

1. *Convolutional layer* berfungsi mengekstraksi fitur seperti tepi dan pola dari *input* dengan menggunakan filter (kernel) yang menghasilkan peta fitur. *Padding* dan *stride* digunakan untuk mengatur ukuran *output*.

$$Output = \frac{inputsize + 2 \times padding - kernel}{stride} + 1 \quad (2.1)$$

2. *Batch normalization layer* berfungsi menormalkan *output* dengan *mean* 0 dan *variance* 1, mempercepat pelatihan, meningkatkan stabilitas, dan memungkinkan pembelajaran parameter skala (γ) dan pergeseran (β).
3. *ReLU layer* berfungsi mengubah nilai negatif menjadi nol, menjaga nilai positif, dan memperkenalkan non-linearitas untuk mempelajari hubungan kompleks dalam data.
4. *Max pooling layer* berfungsi mengambil nilai maksimum dalam area tertentu untuk mengurangi dimensi data, menjaga fitur penting, dan meningkatkan efisiensi model.

$$Output Pooling = \frac{inputsize - kernel}{stride} + 1 \quad (2.2)$$

5. *Fully connected layer* berfungsi menggabungkan semua fitur yang telah diekstraksi untuk menghasilkan *output* akhir, seperti skor atau prediksi kelas.
6. *Softmax layer* berfungsi mengubah skor menjadi probabilitas antar kelas dengan memastikan total probabilitas adalah 1.
7. *Classification layer* berfungsi menghitung loss untuk mengukur kesalahan prediksi, digunakan untuk memperbaiki model selama pelatihan.

Pada CNN terdapat juga parameter pelatihan yang dapat diatur sesuai kebutuhan, seperti *learning rate*, *epoch*, *validation frequency*, *batch size*, dan *literacy*.

1. *Learning rate* adalah parameter yang menentukan seberapa besar langkah pembaruan bobot pada setiap iterasi pelatihan. Nilai *learning rate* yang terlalu

besar dapat menyebabkan model melewati nilai optimal dan gagal mencapai konvergensi. Sebaliknya, nilai yang terlalu kecil membuat pelatihan lambat dan bisa terjebak pada minimum lokal. *Learning rate* memiliki rentang nilai dari 0 hingga 1, biasanya, nilai *learning rate* dipilih dalam rentang 10^{-1} hingga 10^{-6} tergantung pada *database* dan model yang digunakan.

2. *Batch size* adalah jumlah sampel data yang diproses sekaligus sebelum memperbarui bobot. Batch kecil (*mini-batch*) memungkinkan pembaruan bobot lebih sering, namun cenderung lebih berisik. *Batch* besar memberikan pembaruan lebih stabil, tetapi membutuhkan lebih banyak memori dan dapat memperlambat iterasi. Nilai *batch size* yang umum digunakan adalah 32, 64, atau 128.
3. *Epoch* adalah satu siklus penuh pelatihan di mana seluruh *database* digunakan untuk memperbarui bobot model. Semakin banyak *epoch*, semakin lama model dilatih, sehingga memungkinkan model belajar lebih baik. Namun, jika jumlah *epoch* terlalu besar, model berisiko overfitting terhadap data pelatihan.
4. *Validation frequency* adalah seberapa sering model divalidasi menggunakan data validasi selama pelatihan. Frekuensi yang lebih tinggi memberikan umpan balik lebih cepat tentang performa model, tetapi meningkatkan waktu komputasi. Sebaliknya, frekuensi yang rendah mengurangi overhead tetapi bisa melewatkan pola kesalahan yang terjadi di awal pelatihan.
5. *Iteration* adalah jumlah langkah atau *batch* yang diperlukan untuk menyelesaikan satu *epoch*.

CNN bekerja dengan cara mengekstraksi fitur dari gambar *input* dan secara bertahap menyederhanakannya hingga menghasilkan klasifikasi atau prediksi yang akurat. Prinsip kerja CNN terdiri dari beberapa tahapan utama:

1. *Input* Gambar, gambar yang dimasukkan ke dalam CNN dalam bentuk matriks nilai piksel, gambar berwarna memiliki tiga saluran (RGB) yang mewakili intensitas warna merah, hijau, dan biru.

2. *Convolution Layer* (Lapisan Konvolusi), bekerja dengan filter (kernel) yang berupa matriks kecil bergerak di atas gambar untuk mengekstrak fitur seperti tepi, sudut, tekstur, dan pola bentuk. Hasil dari operasi ini disebut *feature map*, yang merepresentasikan fitur penting dari gambar.
3. *Activation Function* (ReLU) digunakan untuk menambahkan *non-linearitas* pada jaringan agar model dapat menangkap hubungan yang kompleks dalam gambar. Fungsi ini menggantikan nilai negatif dalam *feature map* dengan nol, sehingga hanya nilai positif yang dipertahankan.
4. *Pooling Layer* (Lapisan *Pooling*) untuk mengurangi dimensi *feature map* dan mempertahankan informasi yang penting, digunakan *pooling layer* seperti *Max Pooling*. *Pooling* mengambil nilai maksimum dalam area kecil dari *feature map*, sehingga ukuran data menjadi lebih kecil tetapi tetap mempertahankan fitur penting.
5. *Fully Connected Layer* (Lapisan Terhubung Penuh), setelah melalui beberapa lapisan konvolusi dan *pooling*, fitur yang telah diekstraksi diratakan (*flattening*) dan dimasukkan ke dalam *fully connected layer*, yang berfungsi sebagai lapisan klasifikasi. Lapisan ini bekerja seperti jaringan saraf tiruan klasik (*Multilayer Perceptron/MLP*) dan menggunakan *Softmax* untuk menghasilkan probabilitas prediksi kelas.
6. *Training* dan *Backpropagation*, CNN dilatih menggunakan algoritma *backpropagation* dan optimasi (*Stochastic Gradient Descent*) untuk memperbarui bobot filter, sehingga jaringan dapat belajar dari kesalahan dan meningkatkan akurasi seiring waktu.

2.2.5 Fuzzy Logic

Logika *Fuzzy* merupakan suatu logika yang memiliki nilai keaburan atau kesamaran (*fuzzyness*) antara benar atau salah [21]. Logika *fuzzy* memiliki nilai pada rentang 0 hingga 1 yang dapat digunakan untuk menterjemahkan suatu besaran yang diekspresikan menggunakan bahasa (*linguistic*). Logika *fuzzy* dapat diimplementasikan pada penggabungan antara *output* hasil deteksi warna dan deteksi bentuk, sehingga menghasilkan satu *output* yang dapat disesuaikan. Logika

Fuzzy Mamdani adalah salah satu metode dalam sistem logika *fuzzy* yang digunakan untuk pengambilan keputusan berdasarkan aturan-aturan yang bersifat *if-then*. Metode ini diperkenalkan oleh Ebrahim Mamdani pada tahun 1975 untuk mengontrol sistem kompleks dengan pendekatan yang menyerupai pemikiran manusia. Berikut adalah elemen utama dalam logika *fuzzy* Mamdani:

- a. Fuzzifikasi, *input* crisp (tegas) dari sistem diubah menjadi nilai *fuzzy* menggunakan fungsi keanggotaan yang sesuai dengan himpunan *fuzzy*.
- b. Basis Aturan (Rule Base), menggunakan aturan berbentuk *if-then*, seperti: “*If Area Besar and Eksentrisitas Elips, then Jenis Buah Alpukat*” aturan ini dinyatakan dalam bentuk himpunan *fuzzy*.
- c. Inferensi *Fuzzy (Fuzzy Inference)*, menggunakan operator logika *fuzzy* (seperti AND, OR, NOT) untuk menentukan derajat kebenaran *output fuzzy* berdasarkan aturan-aturan yang ada.
- d. Defuzzifikasi, hasil berupa himpunan *fuzzy* dari proses inferensi diubah kembali menjadi nilai crisp menggunakan metode centroid untuk mendapatkan *output* final. Defuzzifikasi *centroid* atau *Center of Gravity (COG)* adalah metode defuzzifikasi yang menentukan nilai crisp dengan mencari titik pusat dari area fungsi keanggotaan *fuzzy*. Metode ini menghitung rata-rata tertimbang dari semua nilai dalam domain keluaran, di mana bobotnya ditentukan oleh derajat keanggotaan masing-masing nilai.

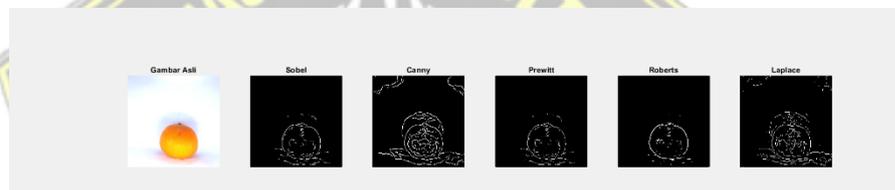
2.2.6 Deteksi Objek Berdasarkan Warna dan Bentuk

Deteksi objek dapat melalui citra warna dan bentuk diawali dengan pengenalan sebuah pola. Pola merupakan bentuk/wujud yang dapat diidentifikasi dan dapat diartikan dari ciri-ciri yang ada padanya (*features*) [18]. Ciri-ciri yang didapatkan bertujuan sebagai pembeda antara sebuah pola dengan pola yang lain. Suatu ciri-ciri yang dikatakan baik ialah ciri yang memiliki hal pembeda yang beragam dan juga unik, sehingga pola yang terbentuk dari ciri tersebut mempunyai ketepatan yang tinggi. Suatu pola mendapatkan ciri-cirinya dari hasil penilaian terhadap objek yang diuji.

Warna dan bentuk adalah dua fitur utama yang sering digunakan dalam pengenalan objek. Warna dapat dengan mudah diukur dalam ruang warna seperti RGB atau HSV, dan sering digunakan untuk segmentasi gambar [19].

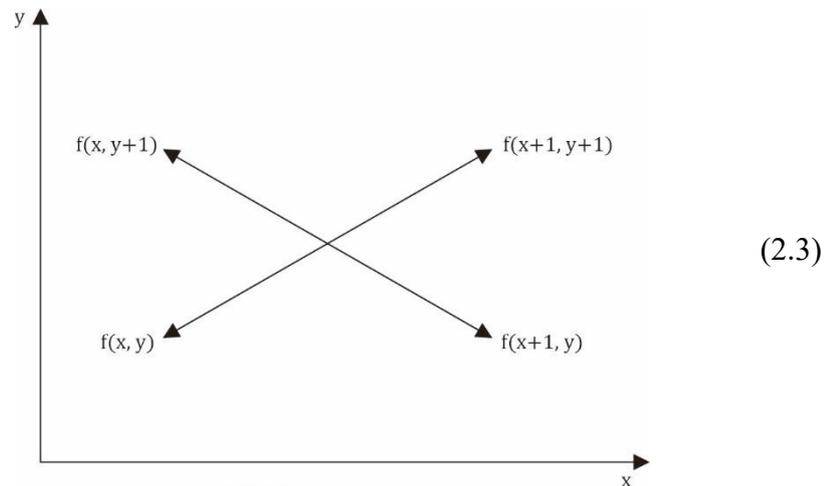
a. Deteksi Tepi (*Edge Detection*)

Bentuk objek dapat diekstraksi menggunakan teknik edge detection (deteksi tepi). Tepi adalah perubahan nilai intensitas derajat keabuan yang mendadak besar dalam jarak yang singkat. Tepi menggambarkan suatu batas-batas objek dan karena itu tepi berguna untuk dilakukan proses segmentasi dan identifikasi objek dalam citra. Terdapat beberapa jenis deteksi tepi yang sering digunakan, antara lain: sobel, canny, prewit, robert dan lain-lain [11].



Gambar 2. 5. Jenis-jenis metode edge detection

Dilihat pada gambar maka operator Roberts cocok untuk melakukan deteksi tepi pada buah dikarenakan noise yang cukup sedikit dibandingkan operator lain. Operator deteksi tepi Roberts atau juga dikenal dengan operator silang. Operator ini berpegang pada gradien dengan menggunakan dua kernel yang mempunyai ukuran 2x2 pixel [20].A



Roberts bekerja dengan menerapkan dua kernel (filter) 2x2 pada citra untuk menghitung gradien intensitas dalam dua arah diagonal, yaitu:

Kernel untuk arah diagonal utama:

$$G_x = \begin{bmatrix} +1 & 0 \\ 0 & -1 \end{bmatrix} \quad (2.4)$$

Kernel untuk arah diagonal sekunder:

$$G_y = \begin{bmatrix} 0 & +1 \\ -1 & 0 \end{bmatrix} \quad (2.5)$$

Gradien gabungan dihitung sebagai:

$$G = \sqrt{G_x^2 + G_y^2} \quad (2.6)$$

Proses pendeteksian tepi pada operator Roberts terdapat 3 langkah yaitu konvolusi citra dengan 2 kernel G_x dan G_y , menghitung magnitude gradien (G) untuk mendapatkan intensitas tepi, menerapkan ambang batas untuk menghasilkan citra biner tepi.

b. Eksentrisitas

Eksentrisitas adalah ukuran yang menunjukkan seberapa bulat atau lonjong bentuk suatu objek. Nilai eksentrisitas dihitung berdasarkan hubungan antara sumbu utama dan sumbu minor dari elips yang melingkupi objek tersebut.

Dengan karakteristik $E = 0$ maka objek berbentuk lingkaran sempurna, $0 < E < 1$ maka objek berbentuk elips, dan $E = 1$ maka objek sangat lonjong (degenerasi menjadi lurus).

$$E = \sqrt{1 - \left(\frac{b}{a}\right)^2} \quad (2.7)$$

Keterangan:

- a : Panjang sumbu utama (major axis)
- b : Panjang sumbu minor (minor axis)

c. Luas atau Area

Area adalah ukuran total luas permukaan suatu objek dalam domain gambar. Dalam konteks pengolahan citra digital, area dihitung berdasarkan jumlah piksel yang membentuk objek tersebut.

$$A = \sum_{i,j} f(i,j) \quad (2.8)$$

Keterangan:

$f(i,j)$ nilai biner dari objek pada koordinat (i,j) , dimana:

- $f(i,j) = 1$ adalah piksel bagian objek
- $f(i,j) = 0$ adalah piksel bagian latar belakang

2.2.7 MATLAB Sebagai Alat Pengembangan

MATLAB adalah platform komputasi numerik yang kuat, yang menyediakan berbagai fungsi dan *toolbox* untuk pengembangan dan pelatihan model jaringan syaraf tiruan. *Deep Learning Toolbox* dalam MATLAB memungkinkan pengguna untuk membangun, melatih, dan menguji model CNN dengan mudah, serta mendukung penggunaan GPU untuk akselerasi komputasi [21]. MATLAB juga menyediakan alat visualisasi yang berguna untuk menganalisis performa model dan memahami bagaimana model tersebut bekerja. *Toolbox* yang memungkinkan untuk digunakan dalam mendeteksi sebuah citra adalah *deep learning*, *image processing*,

statistic and machine learning, bahkan *fuzzy logic*. Berikut adalah penjelasan masing masing *toolbox* tersebut:

1. *Deep Learning Toolbox*

Toolbox ini digunakan untuk membangun, melatih, dan menyebarkan jaringan saraf tiruan serta model pembelajaran mendalam. *Toolbox* ini menyediakan arsitektur modern seperti *Convolutional Neural Networks (CNN)*. Fitur utamanya meliputi kemampuan pelatihan menggunakan GPU, fungsi-fungsi *pre-built* untuk membangun *layer* jaringan, serta integrasi dengan *database* besar. *Deep Learning Toolbox* sangat cocok untuk tugas seperti pengenalan gambar.

2. *Image Processing Toolbox*

Toolbox ini dirancang untuk analisis dan pengolahan gambar. Pada *toolbox* ini dapat melakukan operasi seperti deteksi tepi, segmentasi gambar, pengolahan warna, transformasi geometris, dan ekstraksi fitur. *Toolbox* ini mendukung visualisasi gambar serta manipulasi piksel untuk kebutuhan pra-pemrosesan. Ini sangat berguna untuk aplikasi berbasis visi komputer, seperti pengenalan objek dan pengolahan citra.

3. *Statistics and Machine Learning Toolbox*

Toolbox ini menyediakan alat untuk analisis data, pemodelan statistik, dan pengembangan model pembelajaran mesin. Dengan *toolbox* ini dapat melakukan klasifikasi, regresi, clustering, serta pengujian hipotesis. Fitur lain mencakup teknik pengurangan dimensi seperti PCA (*Principal Component Analysis*), validasi model, dan evaluasi kinerja. *Toolbox* ini cocok untuk aplikasi di bidang seperti analisis data dan prediksi.

4. *Fuzzy Logic Toolbox*

Toolbox Fuzzy Logic adalah fitur yang digunakan untuk merancang, mengimplementasikan, dan menguji sistem logika *fuzzy* secara interaktif maupun melalui pemrograman. *Toolbox* ini menyediakan antarmuka grafis

(*Fuzzy Inference System Editor*) untuk mendefinisikan variabel *input-output*, fungsi keanggotaan, aturan *fuzzy*, serta metode inferensi dan defuzzifikasi. MATLAB mendukung dua jenis sistem *fuzzy*, yaitu Mamdani dan Sugeno, yang dapat digunakan sesuai dengan kebutuhan aplikasi. Selain itu, *toolbox* ini memungkinkan integrasi dengan metode optimasi dan pembelajaran mesin, seperti ANFIS (*Adaptive Neuro-Fuzzy Inference System*), untuk meningkatkan akurasi sistem *fuzzy*.

Keunggulan MATLAB selain pada *toolbox* yang disediakan adalah memiliki fitur untuk membuat *interface* sederhana, fitur tersebut dinamakan *Apps Designer*. *Apps Designer* pada MATLAB adalah antarmuka grafis yang memungkinkan pengguna berinteraksi dengan program secara intuitif melalui elemen visual seperti tombol, *slider*, menu, dan grafik, tanpa harus menggunakan perintah di *Command Window*. *Apps Designer* sangat berguna untuk membuat aplikasi interaktif yang dapat digunakan oleh pengguna dengan sedikit atau tanpa pengalaman pemrograman.

2.2.8 Python sebagai Penghubung

Python adalah bahasa pemrograman tingkat tinggi yang sangat serbaguna, dikenal karena sintaksnya yang sederhana, sehingga mudah dipahami dan digunakan oleh pemula maupun profesional. Bahasa ini memiliki ekosistem pustaka yang kaya, menjadikannya pilihan utama untuk berbagai aplikasi, termasuk pembelajaran mesin (*Machine Learning*) dan pengolahan citra (*Image Processing*). Salah satu keunggulan Python adalah kemampuannya untuk berintegrasi dengan berbagai perangkat lunak, termasuk MATLAB. Integrasi ini memungkinkan pengguna memanfaatkan kekuatan Python untuk pengumpulan data atau pra-pemrosesan, sementara MATLAB digunakan untuk analisis mendalam atau simulasi teknis.

Ketika menghubungkan Python dengan MATLAB untuk menangkap data dari kamera, Python bertindak sebagai perantara untuk mengakses perangkat keras seperti webcam atau kamera eksternal. Proses ini sering menggunakan pustaka

Python seperti *OpenCV* untuk menangkap video atau gambar secara *real-time*, yang kemudian dapat diolah lebih lanjut di MATLAB. Penggunaan Python dalam skenario ini menawarkan fleksibilitas tinggi, termasuk kompatibilitas dengan berbagai perangkat dan protokol komunikasi.

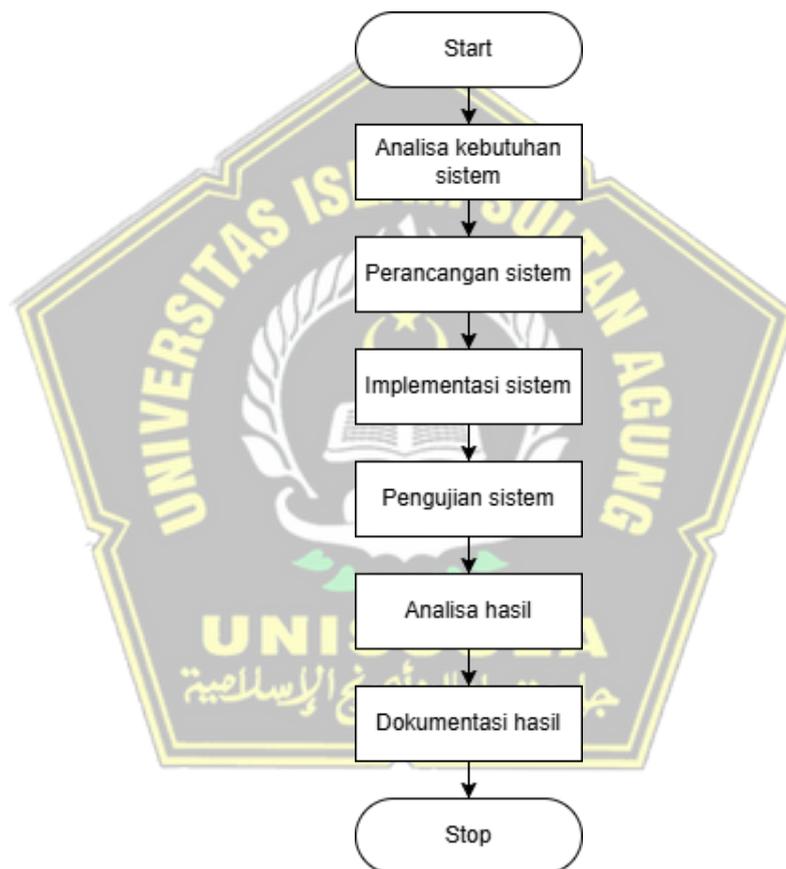
Untuk aplikasi spesifik yang melibatkan penggunaan kamera dari smartphone, aplikasi DroidCam dapat digunakan sebagai jembatan antara perangkat Android dengan Python melalui koneksi WiFi. DroidCam berfungsi mengubah smartphone menjadi kamera IP (Internet Protocol Camera), yang dapat diakses melalui jaringan lokal. Aplikasi ini perlu diinstal baik di smartphone maupun di PC/laptop. Di sisi PC, Python dapat mengakses feed video dari DroidCam melalui IP yang diberikan oleh aplikasi dalam format protokol RTSP. Hal ini memungkinkan penggunaan smartphone sebagai sumber video tanpa perlu kabel fisik, memberikan fleksibilitas dalam pengaturan kamera.

Setelah data video atau gambar ditangkap menggunakan Python, data tersebut dapat disimpan ke dalam format yang kompatibel, seperti file gambar (e.g., PNG, JPEG) atau file video (e.g., MP4). Selanjutnya, MATLAB dapat membaca file ini menggunakan fungsi seperti `imread` atau `VideoReader` untuk analisis atau integrasi lebih lanjut. Proses ini memanfaatkan kekuatan masing-masing platform: Python untuk interfacing perangkat keras dan pengumpulan data, sementara MATLAB untuk pemrosesan data yang lebih canggih, seperti analisis jaringan saraf tiruan (CNN) atau ekstraksi fitur berbasis gambar. Integrasi ini menunjukkan sinergi yang efektif antara fleksibilitas Python dan kapabilitas teknis MATLAB.

BAB III METODE PENELITIAN

3.1. Metode Penelitian

Metode penelitian yang digunakan dalam penelitian ini adalah metode eksperimental research, yang melibatkan pengujian hipotesis melalui pengumpulan data dan analisis empiris untuk mendapatkan hasil yang valid dan reliabel.



Gambar 3. 1 Flowchart metode penelitian

Berdasarkan gambar, maka rancangan yang akan dilakukan pada penelitian ini adalah:

1. Analisis kebutuhan sistem

Sistem ini membutuhkan perangkat keras berupa PC atau laptop, kamera handphone/webcam untuk menangkap gambar, serta penyimpanan untuk menyimpan data hasil deteksi. Perangkat lunak yang digunakan

meliputi Python untuk menangkap gambar dari webcam, MATLAB untuk pengolahan citra dan deteksi menggunakan *neural network*, serta integrasi antara keduanya menggunakan file pemicu. Sistem juga dirancang untuk mampu melakukan deteksi warna dan bentuk secara otomatis serta menampilkan hasilnya dalam *Apps Designer* yang interaktif.

2. Perancangan sistem

Terdiri dari rancangan hardware dan software:

- a. Perancangan Hardware : background, pencahayaan, dan device.
- b. Perancangan Software : manajemen file, pengolahan data, dan interface.

3. Implementasi sistem

Membangun dua bagian utama yaitu script Python dan script MATLAB beserta *interface* pada *Apps Designer*.

4. Pengujian sistem

Sistem diuji secara bertahap, dengan parameter akurasi mendeteksi berbagai jenis buah, mengevaluasi kecepatan *training progress* pada sistem, dan mengevaluasi kecepatan sistem dalam memberikan hasil deteksi.

5. Analisa hasil pengujian

Hasil pengujian dianalisis dan dibandingkan dengan tujuan awal sistem untuk memastikan kesesuaiannya.

6. Dokumentasi hasil

Dokumentasi dilakukan dalam bentuk laporan tertulis yang mencakup analisis hasil pengujian, tabel, grafik, dan dokumentasi kode dari Python dan MATLAB. Panduan pengguna disusun untuk mempermudah penggunaan sistem.

3.2. Analisis Kebutuhan Sistem

Untuk memenuhi kebutuhan sistem secara keseluruhan, akan diuraikan beberapa Analisa kebutuhan sistem yang dilakukan pada penelitian.

3.2.1 Kebutuhan Fungsional Sistem

Sistem dapat menangkap gambar dari kamera handphone dengan rasio 1:1 secara otomatis berdasarkan permintaan. Sistem mampu mendeteksi jenis buah

berdasarkan warna dan bentuk (eksentrisitas dan luas) menggunakan jaringan saraf tiruan. Sistem mampu memutuskan hasil deteksi menjadi satu dengan menghitung berdasarkan hasil deteksi warna dan bentuk yang diproses dengan logika *fuzzy*. Sistem menampilkan hasil deteksi pada antarmuka pengguna (*Apps Designer*). Sistem mendukung pengulangan proses deteksi melalui kontrol *Apps Designer*.

3.2.2 Kebutuhan Non-Fungsional Sistem

Proses pengolahan gambar harus berjalan secara *real-time* tanpa jeda yang signifikan. Antarmuka pengguna dirancang sederhana dan mudah digunakan. Akurasi deteksi mencapai minimal 90% berdasarkan pengujian *database* yang ada. Sistem harus stabil saat digunakan dalam durasi panjang.

3.3. Rancangan Perangkat Keras

Sistem menggunakan kamera handphone untuk meningkatkan kualitas gambar dan menyesuaikan proses data dengan lebih baik. Studio *mini* dirancang secara khusus untuk menghilangkan faktor-faktor lingkungan yang dapat memengaruhi hasil deteksi, seperti bayangan atau pencahayaan yang tidak merata. Studio ini memiliki latar belakang putih untuk meminimalkan gangguan visual dan dilengkapi dengan pencahayaan terkontrol menggunakan lampu led. Kamera handphone ditempatkan pada posisi tetap menghadap objek di dalam studio untuk menangkap gambar dengan sudut pandang konsisten. Device pemrosesan menggunakan Laptop/PC dengan spesifikasi yang mendukung untuk pengolahan data.



Gambar 3. 2 Diagram blok perancangan hardware

Keterangan:

- Objek
Berupa buah yang dideteksi dengan jarak 15cm – 20cm dari lensa kamera. Objek diletakkan pada studio *mini*.

2. Kamera Smartphone

Spesifikasi:

- Resolusi : 50MP
- Aperture : f1.8
- Rasio : 1:3 (720 x 480)
- OS : Android

3. PC/Laptop

Spesifikasi:

- Processor : AMD Ryzen 7 3700U 2.3 ~ 4.0 GHz
- RAM : 16GB
- GPU : AMD Radeon RX Vega 10
- OS : Windows 11 64-bit

4. Monitor

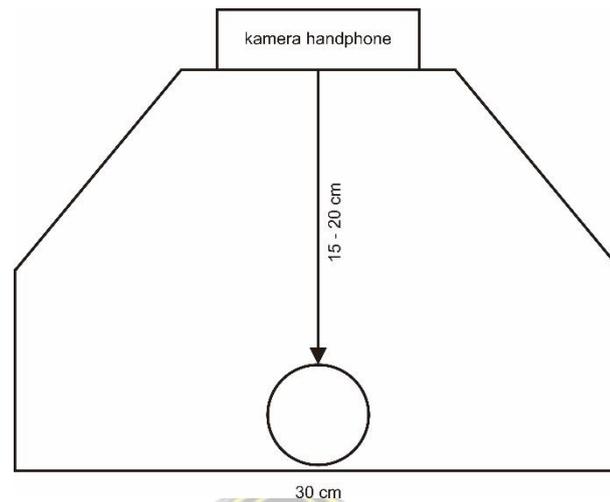
Merupakan komponen display bawaan dari PC dengan spesifikasi:

- Resolusi : FHD 1920 x 1080
- Ukuran : 15 inch
- Refresh rate : 60Hz

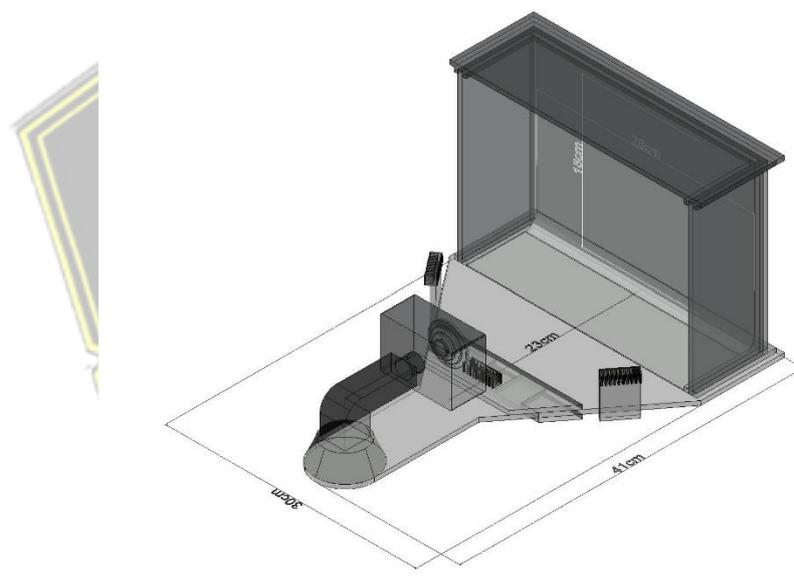
5. Studio *Mini*

Spesifikasi:

- Desain : Berbentuk kotak dengan latar belakang putih.
- Pencahayaan : LED (3 didepan objek, 2 disamping objek, dan 3 diatas objek)
- Dimensi : 41 cm x 30 cm x 20 cm



Gambar 3. 3 Rancangan kamera dan objek



Gambar 3. 4 Rancangan studio *mini*

Keterangan:

Pada gambar terlihat dengan jelas konfigurasi dari studio *mini* yang dirancang untuk mendukung proses deteksi buah. Studio ini dilengkapi dengan handphone dengan kamera belakang menghadap ke objek (buah), handphone dipasang pada tripod untuk memastikan kestabilan dan posisi yang tepat dalam menangkap gambar. Dimensi studio *mini* diatur untuk mendeteksi buah yang memiliki ukuran sedang. Berikut ini adalah alur kerja perangkat keras:

1. Smartphone menangkap gambar buah dalam studio *mini*
2. Data gambar dikirim ke PC melalui koneksi WiFi
3. PC memproses gambar dengan MATLAB dan Python
4. Monitor menampilkan interface sistem

3.4. Rancangan Perangkat Lunak

Perancangan perangkat lunak pada penelitian ini bertujuan untuk menciptakan sistem deteksi buah berdasarkan warna dan bentuk menggunakan metode *Convolutional Neural Network* (CNN). Perangkat lunak dirancang untuk mendukung proses pengolahan citra dari akuisisi gambar hingga pengambilan keputusan. Berikut adalah rincian rancangan perangkat lunak:

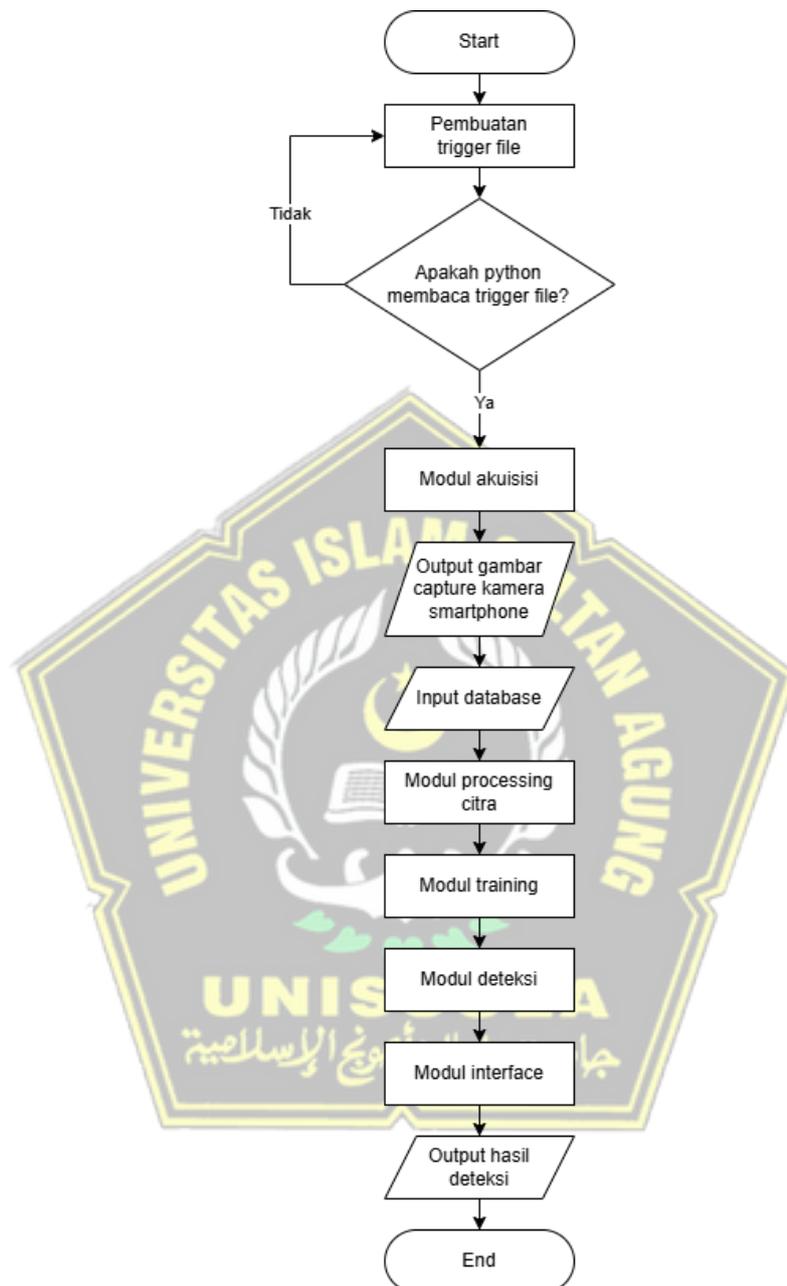
3.4.1 Platform

Perancangan sistem perangkat lunak membutuhkan beberapa aplikasi yang terinstal pada smartphone dan juga PC/Laptop, pada PC yaitu:

1. MATLAB versi R22a
2. VsCode versi 1.95.3
3. Python versi 3.13.0
4. Droidcam Windows Client versi 6.5.2

Pada smartphone hanya satu aplikasi saja, yaitu Droidcam Classic versi 6.27 (Android).

3.4.2 Modul Sistem



Gambar 3. 5. Flowchart keseluruhan sistem

Keseluruhan sistem dimulai dengan menghubungkan semua perangkat keras baik dengan kabel maupun secara nirkabel. Dan menjalankan semua platform dan memastikan semuanya berjalan dengan baik. Sistem diawali dengan pembuatan trigger file yang nantinya akan dibaca oleh Python untuk memulai proses akuisisi gambar. Proses akuisisi gambar menghasilkan gambar dengan ekstensi .jpg yang

akan melewati modul processing citra lalu disimpan pada direktori yang sudah ditentukan. Proses selanjutnya yaitu *input* berupa *database* gambar buah digunakan untuk proses *training* warna dan juga bentuk, yang sebelumnya telah melewati modul processing citra. *Output* modul *training* dan modul processing citra menjadi *input* pada modul deteksi. Modul deteksi dijalankan dan menghasilkan *output* kategori jenis buah yang dideteksi. Hasil deteksi akan ditampilkan pada interface buatan MATLAB yaitu *Apps Designer*. Setelah ditampilkan maka proses deteksi jenis buah telah selesai. Dari rancangan diatas, dapat ditentukan bahwa perangkat lunak yang dibuat terdiri dari beberapa modul utama, yaitu:

1. Modul Akuisisi Gambar

Mengambil gambar dari kamera handphone yang terhubung ke komputer melalui jaringan wifi menggunakan Python.



Gambar 3. 6 Diagram blok modul akuisisi gambar

Keterangan:

- a. Smartphone terinstal Droidcam Classic terhubung ke jaringan yang sama dengan PC/Laptop yang terinstal Droidcam Windows Client menggunakan IP Address dan Port.



Gambar 3. 7 Tampilan IP address pada Droidcam Classic



Gambar 3. 8 Tampilan *input* IP address pada Droidcam Windows Client

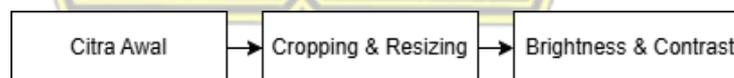
- b. Droidcam mendapatkan akses live kamera smartphone melalui jaringan wifi
- c. Data live kamera diterima dan digunakan Python untuk capture image

```
cap = cv2.VideoCapture(1)
cv2.imshow("Live Camera 1:1", enhanced_frame)
```

2. Modul Processing Citra

Melakukan pengolahan awal pada gambar, seperti cropping, resizing, rotating ke ukuran standar CNN, dan penyesuaian kontras dan kecerahan. Modul terbagi menjadi 2 bagian, yaitu *Preprocessing* dan *Postprocessing*. Pre processing dijalankan setelah akuisisi gambar, sedangkan post processing dijalankan sebelum *training database*, dan sebelum deteksi.

Proses preprocessing pada Python:



Gambar 3. 9 Diagram blok *preprocessing*

Keterangan:

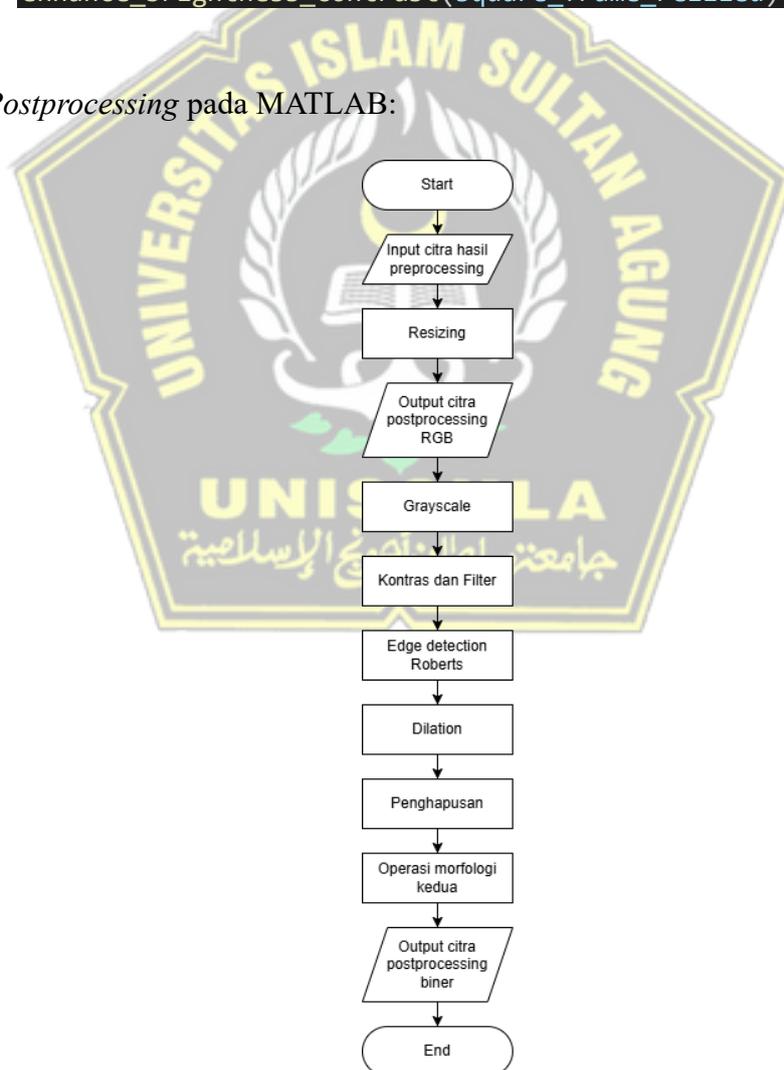
- a. Citra awal berukuran sesuai dengan live kamera dari smartphone yang berukuran 480 x 720
- b. Citra dipangkas dengan ukuran 480 x 480 (1:1) dan ukuran diubah menjadi 200 x 200 piksel sesuai kebutuhan CNN

```
square_frame_resized = cv2.resize(square_frame, (side_length,
side_length))
```

- c. Kecerahan dan kontras pada citra diatur untuk menegaskan citra dan mengurangi intensitas bayangan

```
def enhance_brightness_contrast(frame):
    alpha = 1.5
    beta = 10
    enhanced_frame = cv2.convertScaleAbs(frame, alpha=alpha,
beta=beta)
    return enhanced_frame
enhanced_frame =
enhance_brightness_contrast(square_frame_resized)
```

Postprocessing pada MATLAB:



Gambar 3. 10 Flowchart *postprocessing*

Keterangan:

- a. Citra awal didapat dari *preprocessing* yang telah dilakukan oleh python
- b. Proses resizing dengan *input size* [200 200 3] untuk pelatihan dan deteksi warna, [200 200 1] untuk deteksi bentuk, *postprocessing* untuk warna hanya sampai pada tahap ini

```
inputSize = [200 200 3];
testImageResized = imresize(img, inputSize(1:2));
inputSizeGray = [200 200 1];
testImageResizedGray = imresize(img, inputSizeGray(1:2));
```

- c. Gambar diubah menjadi 1 kernel atau *grayscale* untuk mempercepat proses, dikarenakan edge detection tidak membutuhkan fitur warna

```
imgGray = rgb2gray(img);
```

- d. Kontras diatur untuk memudahkan proses edge detection dan diterapkan median filter untuk mengurangi noise

```
imgContrast = adapthisteq(imgGray);
imgFiltered = medfilt2(imgContrast, [3 3]);
```

- e. Edge detection Roberts dilakukan dengan menghitung gradien dan menentukan ambang batasnya.

```
edgeDetection = edge(imgFiltered, 'roberts');
```

- f. Dilation dilakukan dengan memperluas foreground (piksel putih) pada citra biner dan mengisi lubang kecil pada gambar

```
processedEdges = imdilate(edgeDetection, strel('disk', 3));
processedEdges = imfill(processedEdges, 'holes');
```

- g. Penghapusan yang dimaksud adalah menghapus noise dan objek-objek kecil dibawah nilai tertentu sehingga menyisakan objek utama

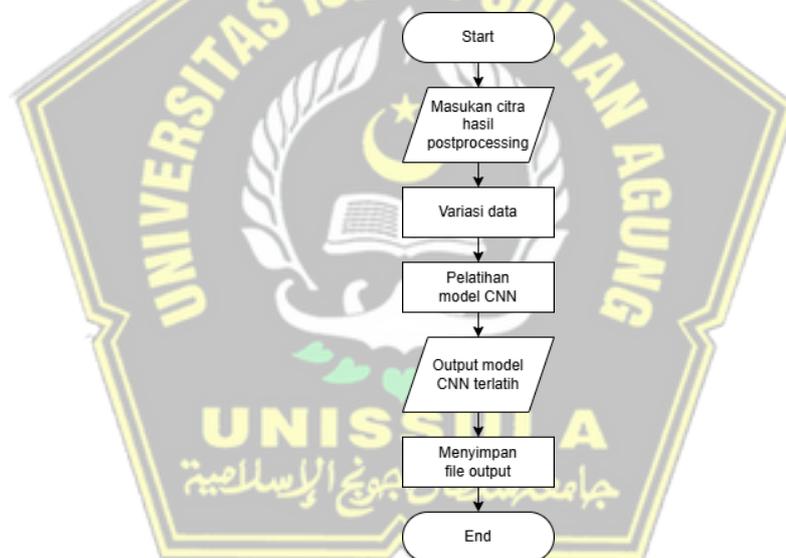
```
filledEdges2nd = bwareaopen(processedEdges, 200);
```

- h. Operasi morfologi (dilation dan penghapusan) dilakukan untuk kedua kalinya untuk memperkuat bentuk objek. Menghasilkan *output* berupa gambar *postprocessing* gray

```
bwImg = imfill(bwImg, 'holes');
bwImg = bwareaopen(bwImg, 100);
se = strel('disk', 5);
bwImg = imclose(imdilate(bwImg, se), se);
```

3. Modul *Training Database*

Modul ini terbagi menjadi *Training Warna* dan *Training Bentuk*. *Training* warna menggunakan CNN untuk melatih sistem dengan *database* yang telah disiapkan. Lalu menyimpan hasil latih tersebut pada folder yang telah ditentukan sehingga dapat digunakan pada modul deteksi.



Gambar 3. 11 Flowchart *training* warna

Keterangan:

- Proses dimulai dengan *input* hasil dari *postprocessing*, citra sudah berukuran 200 x 200 dengan tiga saluran warna sehingga sesuai untuk proses pelatihan
- Variasi data citra dengan membagi menjadi dua kelompok, yaitu data latih sebanyak 80% dan data uji sebanyak 20%. Setiap data akan divariasikan dengan rotasi acak antara -10 derajat hingga 10 derajat dan translasi acak

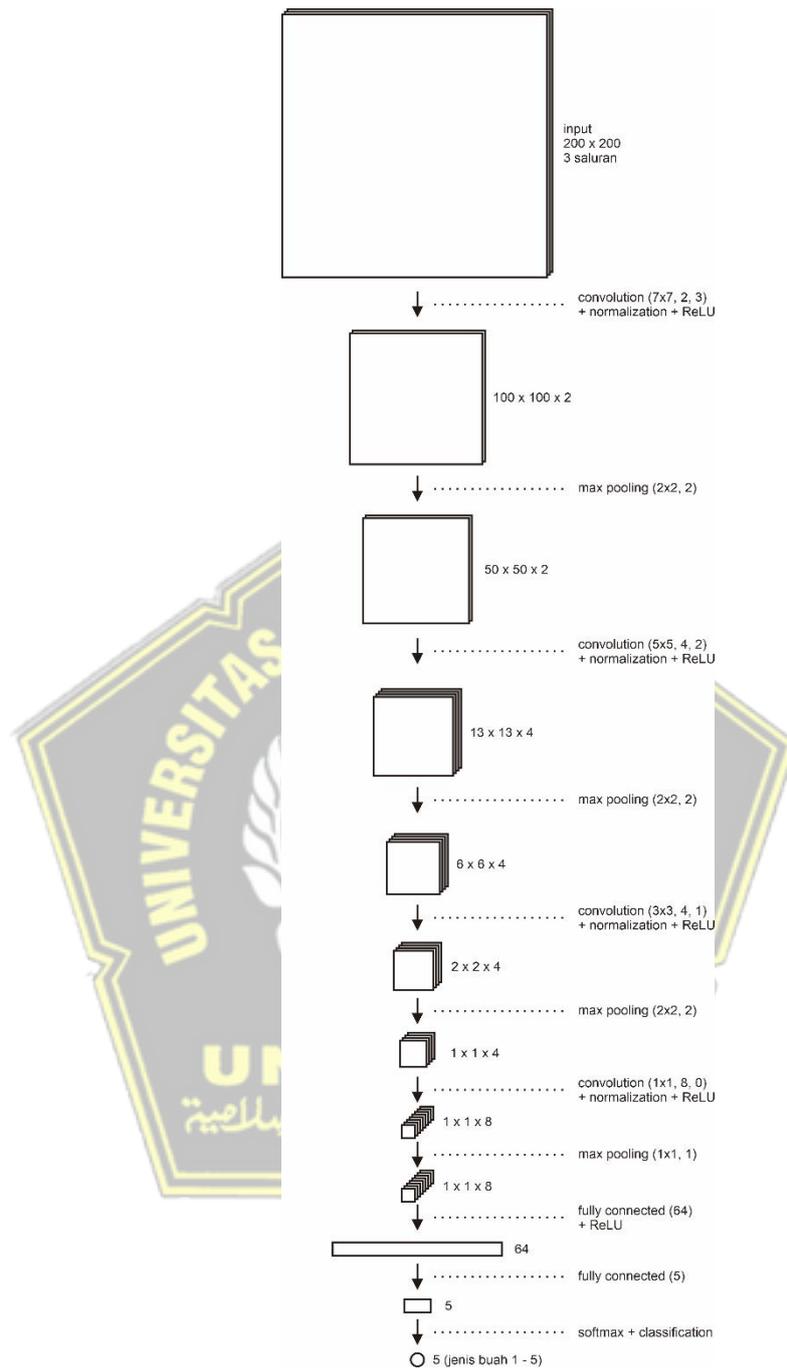
sejauh -3 sampai 3 piksel baik secara *horizontal* maupun *vertical*. Variasi ini berguna untuk meningkatkan generalisasi model.

```
[imdsTrainWarna, imdsValidationWarna] =
splitEachLabel(imdsWarna, 0.8, 'randomized');
imdsTrainWarna.ReadFcn = @(filename)
im2rgb(imreadAndResize(filename, inputSize));
imdsValidationWarna.ReadFcn = @(filename)
im2rgb(imreadAndResize(filename, inputSize));
augmenterWarna = imageDataAugmenter('RandRotation', [-10 10],
'RandXTranslation', [-3 3], 'RandYTranslation', [-3 3]);
augimdsTrainWarna = augmentedImageDatastore(inputSize,
imdsTrainWarna, 'DataAugmentation', augmenterWarna,
'ColorPreprocessing', 'gray2rgb');
augimdsValidationWarna = augmentedImageDatastore(inputSize,
imdsValidationWarna, 'ColorPreprocessing', 'gray2rgb');
```

- c. Pelatihan model CNN dilakukan dengan tiga proses, yaitu *feature* extraction, classification, dan probabilistic distribution. Berikut adalah kode program yang digunakan:

```
layersWarna = [
imageInputLayer(inputSize)
convolution2dLayer(7, 2, 'Padding', 'same')
batchNormalizationLayer
reluLayer
maxPooling2dLayer(2, 'Stride', 2)
convolution2dLayer(5, 4, 'Padding', 'same')
batchNormalizationLayer
reluLayer
maxPooling2dLayer(2, 'Stride', 2)
convolution2dLayer(3, 4, 'Padding', 'same')
batchNormalizationLayer
reluLayer
maxPooling2dLayer(2, 'Stride', 2)
convolution2dLayer(1, 8, 'Padding', 'same')
batchNormalizationLayer
reluLayer
maxPooling2dLayer(1, 'Stride', 1)
fullyConnectedLayer(64)
reluLayer
fullyConnectedLayer(numel(unique(imdsTrainWarna.Labels)))
softmaxLayer
classificationLayer
];
```

Dari kode tersebut dapat digambarkan seperti gambar dibawah ini untuk setiap proses-prosesnya.



Gambar 3. 12 Rancangan arsitektur CNN

Rancangan arsitektur tersusun atas *feature maps* yang terdiri dari empat kali *convolution layer*, *batch normalization layer*, *ReLU layer*, dan *max pooling layer*. Pada bagian *classification* tersusun atas dua kali *fully connected* dan memiliki *ReLU layer* pada perhubung antara keduanya. Bagian terakhir yaitu *probabilistic distribution* yang berisi *softmax* dan

classification menghasilkan lima kategori klasifikasi. Opsi pelatihan juga diatur dengan parameter *learning rate*, *epoch*, *batch size*, *validation frequency*.

```
optionsWarna = trainingOptions('sgdm', ...
    'InitialLearnRate', 0.0001, ...
    'MaxEpochs', 10, ...
    'MiniBatchSize', 32, ...
    'ValidationData', augimdsValidationWarna, ...
    'ValidationFrequency', 10, ...
    'Verbose', false, ...
    'Plots', 'training-progress');
```

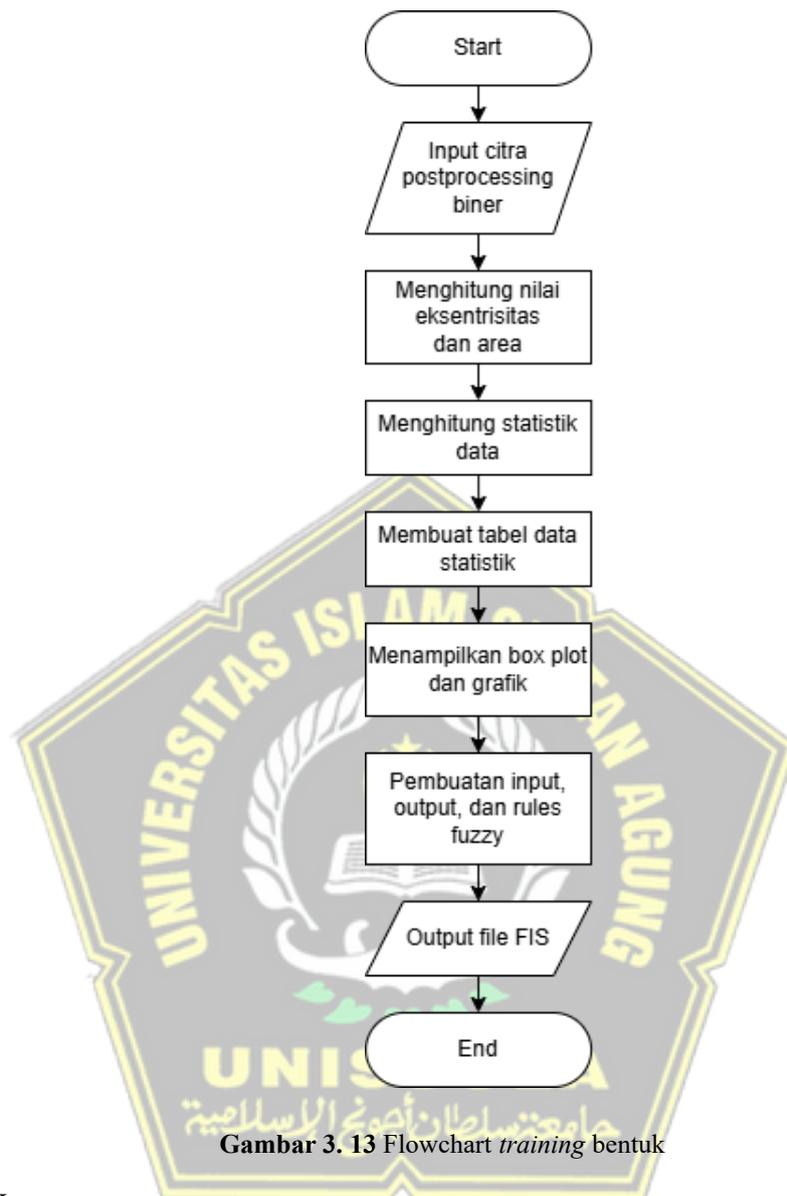
d. Pelatihan CNN menghasilkan *output* berupa train network atau model yang sudah terlatih

```
netWarna = trainNetwork(augimdsTrainWarna, layersWarna,
    optionsWarna);
```

e. Model terlatih disimpan dengan ekstensi *.mat* yang nantinya akan digunakan dalam klasifikasi jenis buah

```
save('fruitCNN_Warna.mat', 'netWarna');
```

Training bentuk menggunakan *database* yang telah diproses menjadi citra biner, citra biner diukur nilai eksentrisitas dan areanya guna untuk mendapatkan sebuah grafik eksentrisitas dan area pada tiap buah. Proses ini melibatkan proses *fuzzy* dan input dari *training* warna.



Keterangan:

- a. Proses dimulai dengan *input* hasil dari *postprocessing* gray, citra sudah berukuran 200 x 200 dengan satu saluran warna (*grayscale*) sehingga sesuai untuk proses pelatihan.
- b. Nilai eksentrisitas dan area untuk masing-masing data diukur

```

stats = regionprops(bwImg, 'Eccentricity', 'Area');
for k = 1:length(stats)
    eccentricities(end+1) = stats(k).Eccentricity;
    areas(end+1) = stats(k).Area;
end
  
```

- c. Nilai tersebut dihitung nilai terendah (min), kuartil pertama (Q1), kuartil kedua (median), kuartil ketiga (Q3), nilai tertinggi, dan rata-ratanya. Hasil perhitungan disimpan dalam sebuah table pada workspace MATLAB dan juga disimpan pada direktori dengan ekstensi .xlsx (file Excel)

```
% Hitung Q1, Median (Q2), Q3, Min, dan Max untuk eccentricities
q1Eccentricities = quantile(currentEccentricities, 0.25);
q2Eccentricities = quantile(currentEccentricities, 0.50);
q3Eccentricities = quantile(currentEccentricities, 0.75);
minEccentricities = min(currentEccentricities);
maxEccentricities = max(currentEccentricities);
avgEccentricities = mean(currentEccentricities);

% Hitung Q1, Median (Q2), Q3, Min, dan Max untuk areas
q1Areas = quantile(currentAreas, 0.25);
q2Areas = quantile(currentAreas, 0.50);
q3Areas = quantile(currentAreas, 0.75);
minAreas = min(currentAreas);
maxAreas = max(currentAreas);
avgAreas = mean(currentAreas);

% Simpan hasil ke dalam tabel
fruitStats{i, 1} = subfolders(i).name;
fruitStats{i, 2} = minEccentricities;
fruitStats{i, 3} = q1Eccentricities;
fruitStats{i, 4} = q2Eccentricities;
fruitStats{i, 5} = q3Eccentricities;
fruitStats{i, 6} = maxEccentricities;
fruitStats{i, 7} = avgEccentricities;
fruitStats{i, 8} = minAreas;
fruitStats{i, 9} = q1Areas;
fruitStats{i, 10} = q2Areas;
fruitStats{i, 11} = q3Areas;
fruitStats{i, 12} = maxAreas;
fruitStats{i, 13} = avgAreas;

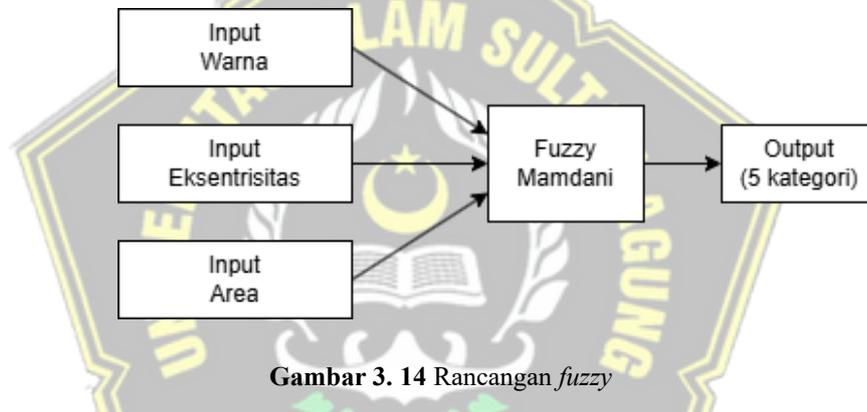
% Konversi ke tabel
fruitStatsTable = cell2table(fruitStats, 'VariableNames', ...
    {'Fruit', 'Min_Eccentricity', 'Q1_Eccentricity',
    'Median_Eccentricity', 'Q3_Eccentricity', 'Max_Eccentricity',
    'Avg_Eccentricity', ...
    'Min_Area', 'Q1_Area', 'Median_Area', 'Q3_Area',
    'Max_Area', 'Avg_Area'});
disp(fruitStatsTable);

% Simpan hasil ke file Excel
writetable(fruitStatsTable,
    'D:\Ihza\TA\System\Matlab\data_statik_buah.xlsx');
disp('Hasil Q1, Median, Q3, Minimum, dan Maksimum disimpan ke
Excel.');
```

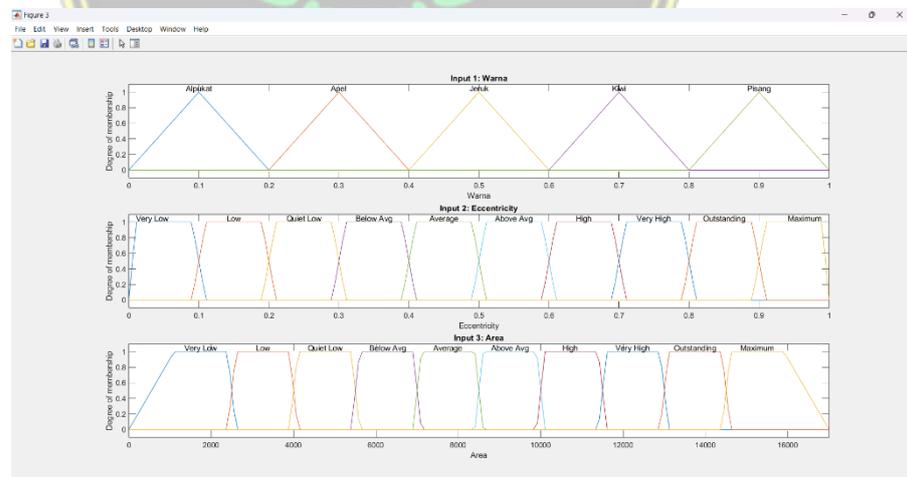
- d. Nilai statistik ditampilkan dalam bentuk box plot dan grafik sebagai visual data yang diperoleh.

```
boxplot(allEccentricities, categories, 'Colors', 'k', 'Widths',
1, 'Symbol', 'r+');
box = boxplot(allAreas, categories, 'Colors', 'k', 'Widths', 1,
'Symbol', 'r+');
```

- e. Pembuatan *input*, *output* dan *rules fuzzy* didasari oleh hasil perhitungan nilai statistic dan juga *output* kategori *training* warna. Kategori *training* warna akan digunakan sebagai *input* warna, statistic data *training* bentuk akan digunakan sebagai pembuatan *rules* pada *fuzzy logic*. Keseluruhan kategori akan menjadi kelas dalam *output fuzzy*.



Gambar 3. 14 Rancangan *fuzzy*



Gambar 3. 15. Rancangan pembagian kelas *input fuzzy*

Kategori *input* warna terbagi sesuai dengan jumlah jenis buah yang dapat dideteksi, pada penelitian ini buah yang dideteksi berjumlah 5. Kategori

input eksentrisitas dan area terbagi menjadi 10 kategori, yaitu *very low*, *low*, *quiet low*, *below average*, *average*, *above average*, *high*, *very high*, *outstanding*, dan *maximal*. *Rules fuzzy* dirancang dengan memperhatikan nilai minimal, kuartil pertama, kuartil ketiga, dan nilai maksimal setiap buah. Apabila buah Alpukat terhitung nilai eksentrisitasnya berada pada kategori *high* dan *very high*, kemudian nilai areanya berada pada kategori *high* hingga *maximal*, maka *rules* yang dibuat akan seperti diagram berikut:



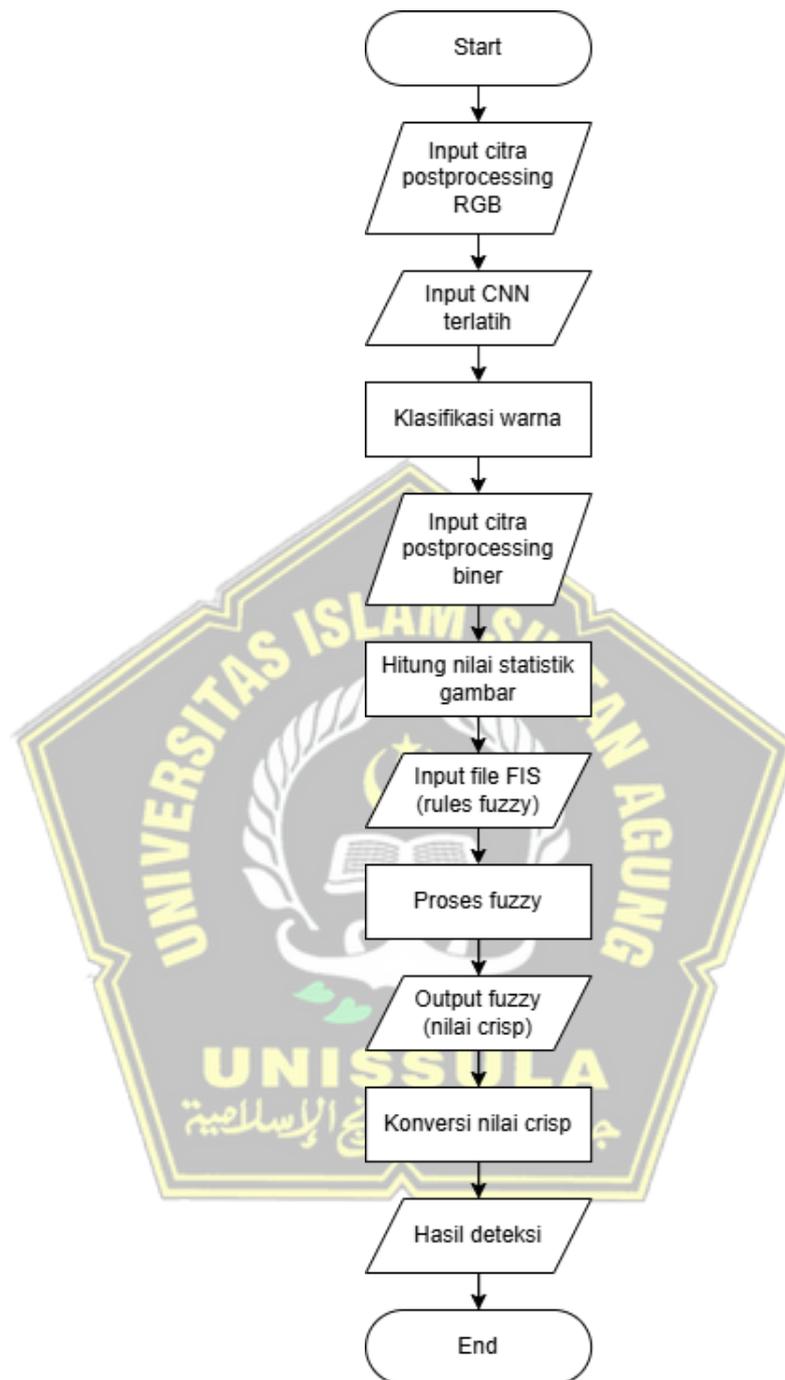
Gambar 3. 16. Rancangan *rules fuzzy*

- f. Sistem *fuzzy* yang sudah dibuat akan disimpan pada direktori dengan ekstensi *.fis* yang nantinya akan digunakan pada modul deteksi

```
writeFIS(fis, 'fuzzyBuah.fis');
```

4. Modul Deteksi Buah

Menggunakan CNN yang telah dilatih untuk mendeteksi jenis buah berdasarkan warna. CNN diimplementasikan menggunakan model yang dimuat dari file MATLAB. Pada bentuk menggunakan nilai statistik bentuk buah yang telah diukur dan diplot dalam grafik. Kedua model (warna dan bentuk) dihubungkan menggunakan logika *fuzzy*.



Gambar 3. 17 Flowchart modul deteksi

Keterangan:

- a. Proses deteksi dimulai dengan *input* sebuah citra dari modul akuisisi yang telah melewati modul processing sehingga terbagi menjadi dua buah citra, yaitu RGB dan biner.

- b. Pada citra RGB dilakukan klasifikasi berdasarkan warna menggunakan CNN yang telah dilatih sebelumnya. Menghasilkan *output* berupa kategori jenis buah (Apel, Alpukat, dan lainnya)

```
loadedNetWarna = load('fruitCNN_Warna.mat').netWarna;
labelWarna = classify(loadedNetWarna, testImageResized);
```

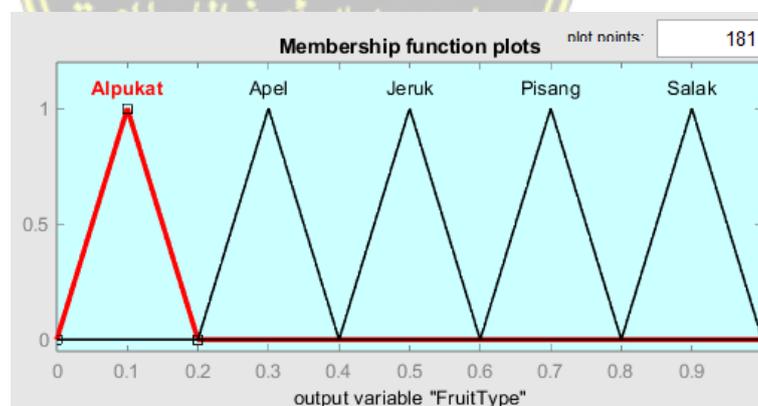
- c. Pada citra biner dilakukan perhitungan nilai statistic (eksentrisitas dan area). Menghasilkan nilai eksentrisitas dan nilai area.

```
stats = regionprops(bwImg, 'Eccentricity', 'Area');
eccentricity = stats.Eccentricity;
area = stats.Area;
```

- d. *Output* hasil deteksi warna dan *output* nilai eksentrisitas serta nilai area digunakan sebagai *input* pada logika *fuzzy*. *Rules* dan sistem pada logika *fuzzy* diambil dari file .fis yang telah dibuat sebelumnya.

```
fisPath = 'D:\Ihza\TA\System\Matlab\fuzzyBuah.fis';
if ~exist(fisPath, 'file')
    error('File FIS tidak ditemukan: %s', fisPath);
end
fuzzyBuah = readfis(fisPath);
inputs = [nilaiWarna, eccentricity, area];
output = evalfis(fuzzyBuah, inputs);
```

- e. Hasil dari proses *fuzzy* berupa nilai crisp rentang 0 hingga 1.



Gambar 3. 18 Rancangan *output fuzzy*

- f. *Output* proses *fuzzy* (nilai crisp) dikonversi sesuai dengan *rules* yang telah diatur pada modul *training*.

```

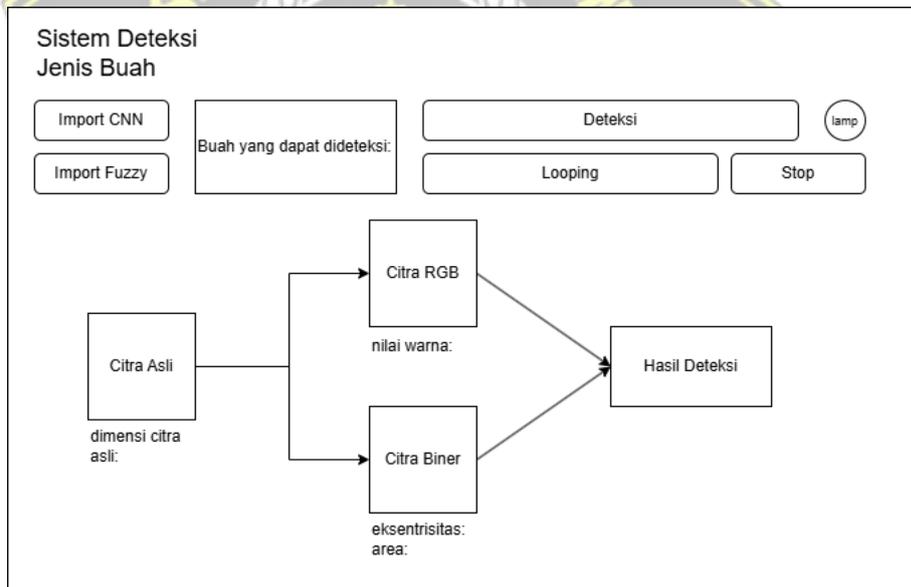
if output >= 0 && output < 0.2
    kategori = subfolders(1).name;
elseif output >= 0.2 && output < 0.4
    kategori = subfolders(2).name;
elseif output >= 0.4 && output <= 0.6
    kategori = subfolders(3).name;
elseif output >= 0.6 && output <= 0.8
    kategori = subfolders(3).name;
elseif output >= 0.8 && output <= 1
    kategori = subfolders(3).name;
else
    kategori = 'Nilai keluar dari rentang';
end

```

g. Hasil deteksi akan ditampilkan sesuai kategori jenis buah, contoh: “Alpukat”. Hasil dapat dijadikan *input* untuk ditampilkan pada *Apps Designer*.

5. Modul Interface

Menampilkan hasil deteksi pada antarmuka pengguna (*Apps Designer*).



Gambar 3. 19. Rancangan *Apps Designer*.

Keterangan:

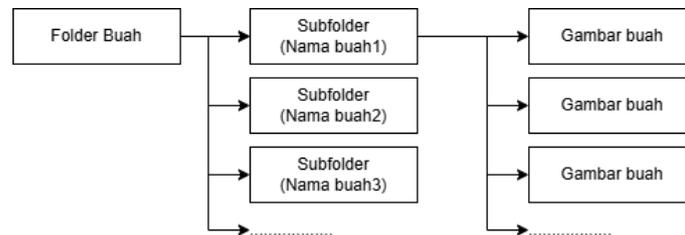
- Tombol import CNN difungsikan untuk memasukkan data latih CNN yang akan digunakan pada proses klasifikasi.

- Tombol import *fuzzy* difungsikan untuk memasukkan *rules fuzzy* yang digunakan untuk memutuskan proses pengambilan keputusan atas hasil deteksi berdasarkan nilai warna, eksentrisitas, dan area.
- Kolom buah yang dapat dideteksi akan menampilkan buah apa saja yang terdapat pada *database* sehingga pengguna dapat mengetahui buah apa saja yang dapat dideteksi.
- Tombol deteksi difungsikan untuk menjalankan proses deteksi buah
- Tombol *looping* difungsikan untuk melakukan deteksi secara berulang sehingga tidak diperlukan untuk klik tombol deteksi secara berulang kali.
- Tombol stop digunakan untuk menghentikan proses *looping*.
- Kolom hasil deteksi akan menampilkan hasil dari deteksi berupa nama jenis buah beserta kecepatan deteksinya.
- Axes1 citra asli menampilkan gambar dalam kondisi original sebelum dilakukan *postprocessing* beserta dimensi ukurannya.
- Axes2 citra RGB menampilkan gambar setelah *preprocessing*, yaitu gambar yang digunakan untuk proses deteksi warna, beserta menampilkan nilai warnanya.
- Axes2 citra biner akan menampilkan gambar setelah dilakukan *postprocessing*, gambar dalam bentuk citra biner beserta nilai eksentrisitas dan area.

3.4.3 Database

Data gambar buah alpukat, apel, jeruk, pisang dan tomat dikumpulkan dari pengambilan gambar secara manual sebanyak 90 gambar pada setiap jenis buahnya, gambar diambil menggunakan kamera smartphone dengan jarak ke permukaan buah 15cm sampai 20cm menyesuaikan ukuran buah, gambar diambil pada kondisi

cahaya yang terkontrol dan berlatar belakang putih. Data gambar diambil telah melewati *preprocessing*. Berikut adalah aturan penyimpanan *database*:



Gambar 3. 20 Rancangan penyusunan folder *database*

Dalam folder *database* terdapat subfolder-subfolder yang berisi gambar gambar buah untuk dilatih. Subfolder diberi nama sesuai jenis buah didalamnya.



Gambar 3. 21 Contoh isi dalam subfolder nama buah Pisang

3.5. Rancangan Pengujian

Pengujian akan dilakukan pada penelitian ini yang akan menguji kinerja sistem baik pada perangkat lunak maupun perangkat keras.

3.5.1 Pengujian Perangkat Keras

Tabel 3.1. Rancangan pengujian perangkat keras

No	Komponen	Rancangan pengujian	Tujuan
1	Kamera smartphone	Menguji kamera smartphone apakah dapat menangkap gambar dengan baik.	Kamera smartphone dapat menangkap gambar dengan baik.
2	Laptop/PC	Menguji laptop dapat menjalankan sistem dengan baik dan lancar pada MATLAB maupun Python	PC/Laptop dapat menjalankan sistem dengan lancar tanpa kendala.
3	Monitor	Menguji monitor bawaan laptop/PC dapat memberikan visual sistem dengan baik	Monitor dapat menampilkan visual sistem dengan baik.

Tabel 3.2. Rancangan pengujian perangkat keras

4	Studio <i>mini</i>	Menguji studio <i>mini</i> untuk melancarkan proses deteksi dengan mencegah adanya bayangan dan membuat brackground berwarna putih untuk memudahkan deteksi warna.	Studio <i>mini</i> dapat mempermudah proses deteksi.
---	--------------------	--	--

3.5.2 Pengujian Perangkat Lunak

Tabel 3. 3. Rancangan pengujian perangkat lunak

No	Komponen	Keterangan komponen	Rancangan pengujian	Tujuan
1	Platform	MATLAB	Menguji MATLAB untuk melakukan proses komputasi dan image processing dengan 47ancer.	MATLAB dapat menjalankan proses dengan lancar.

Tabel 3. 4. Rancangan pengujian perangkat lunak

		VsCode	Menguji VsCode untuk melakukan debugging program python dan menjalankannya.	VsCode dapat melakukan debugging dan menjalankan program python.
		Python	Menguji program python untuk mengakses kamera smartphone melalui Droidcam dan capture gambar serta melakukan image processing.	Python dapat mengakses kamera smartphone dan melakukan sebagian proses sistem yang dibutuhkan
		Droidcam Windows Client	Menguji Droidcam Windows Client untuk menghubungkan kamera smartphone dengan PC melalui WiFi.	Droidcam Windows Client dapat menghubungkan antar kamera smartphone dengan PC

Tabel 3. 5. Rancangan pengujian perangkat lunak

		Droidcam Classic	Menguji Droidcam Classic untuk menampilkan IP address dan port untuk koneksi ke PC.	Droidcam classic dapat menampilkan alamat yang sesuai.
2	Modul sistem	Akuisisi gambar	Menguji modul akuisisi gambar untuk capture gambar dengan perintah dari sistem	Modul dapat mengakuisisi gambar dari kamera <i>smartphone</i>
		<i>Processing</i> citra	Menguji modul <i>processing</i> citra untuk menjalankan <i>preprocessing</i> dan <i>postprocessing</i> yang menghasilkan citra sesuai kebutuhan sistem	Modul dapat melakukan <i>processing</i> citra seperti <i>resize</i> , <i>grayscale</i> , <i>crop</i> , dan lainnya

Tabel 3. 6. Rancangan pengujian perangkat lunak

		<i>Training data</i>	Menguji modul <i>training data</i> untuk menjalankan proses pelatihan dan menghasilkan sistem yang terlatih	Modul dapat menghasilkan sistem yang terlatih
		Proses deteksi	Menguji modul deteksi untuk mendeteksi <i>input</i> gambar dari buah yang dideteksi	Modul dapat mendeteksi citra dengan akurat
		<i>Apps Designer</i>	Menguji <i>Apps Designer</i> untuk membuat interface yang dapat memudahkan pengguna dalam menjalankan sistem	<i>Apps Designer</i> dapat dijalankan dan dapat memudahkan pengguna

Tabel 3. 7. Rancangan pengujian perangkat lunak

3	<i>Database</i>		Menguji <i>database</i> untuk objek pelatihan sistem sehingga meningkatkan akurasi sistem	<i>Database</i> dapat meningkatkan akurasi sistem
---	-----------------	--	---	---



BAB IV HASIL PERANCANGAN DAN PENGUJIAN

4.1. Hasil Perancangan

Berikut ini adalah hasil perancangan sistem deteksi buah otomatis berdasarkan warna dan bentuk menggunakan MATLAB dengan *neural network*.

4.1.1 Hasil Perancangan Perangkat Keras

Sistem deteksi otomatis jenis buah memanfaatkan beberapa perangkat keras untuk meningkatkan kualitas *input* citra yang baik, pemrosesan yang cepat, dan hasil deteksi yang akurat. Berikut ini adalah hasil dari perancangan perangkat keras:



Gambar 4. 1. Perangkat sistem deteksi otomatis jenis buah

Terdiri dari 1 buah PC yang telah sesuai dengan spesifikasi dan terhubung dengan monitor, 1 buah smartphone dengan kamera bawaannya, dan 1 buah set alat studio *mini* beserta pencahayaannya. Dalam hasil perancangan terdapat studio *mini* yang terbuat dari bahan PVC board, dengan dimensi 41cm x 30cm x 20cm. Dan pencahayaan dari LED dengan suhu warna putih netral yang terpasang sesuai rancangan.



Gambar 4. 2. Studio *mini* dengan pencahayaan tidak dinyalakan



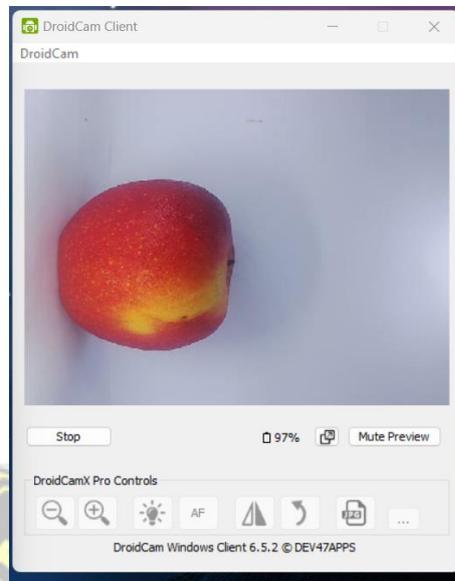
Gambar 4. 3. Studio *mini* dengan pencahayaan dinyalakan

Pada saat pencahayaan dinyalakan dapat dilihat bahwa objek buah tidak memiliki bayangan yang signifikan.

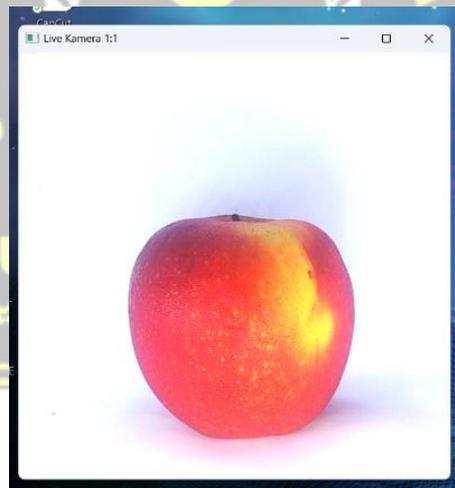
4.1.2 Hasil Perancangan Akuisisi Gambar

Pada bagian akuisisi gambar terdiri atas tampilan pada Droidcam Windows Client dan juga Python. Hasilnya adalah berupa live video yang diambil dari kamera smartphone secara wireless. Dengan perbedaan pada Droidcam Windows Client

video masih berupa original tanpa processing citra, sedangkan pada Python video telah diproses dengan modul processing citra.



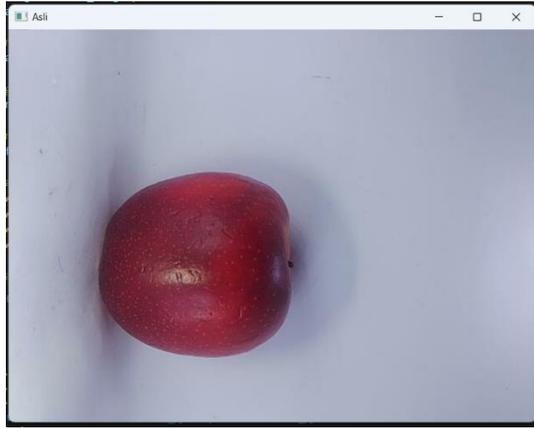
Gambar 4. 4. Live video pada Droidcam Windows Client



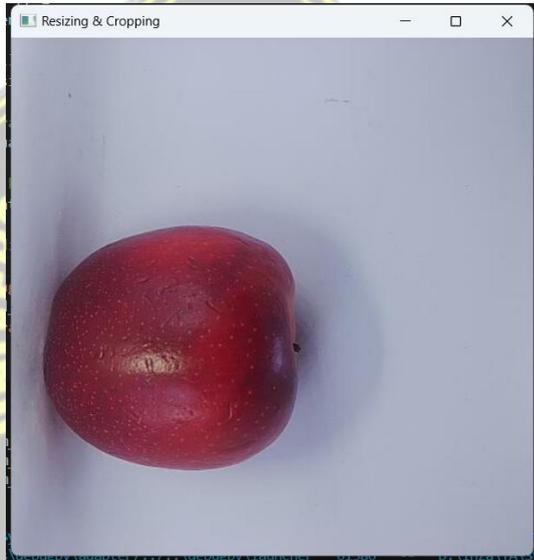
Gambar 4. 5. Live video pada Python

4.1.3 Hasil Perancangan Processing Citra

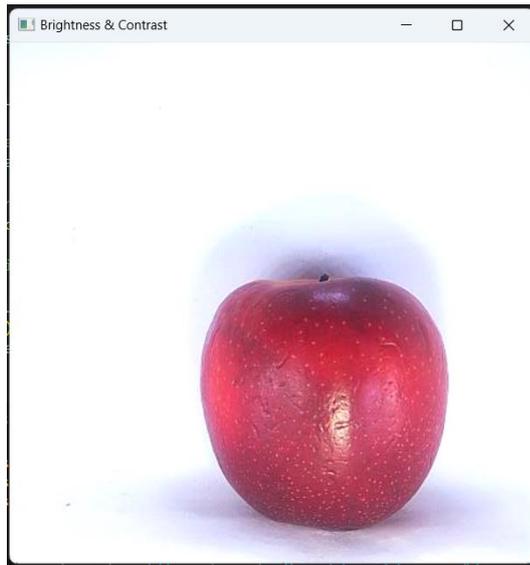
Processing citra berjalan pada dua platform, yaitu MATLAB dan Python. Pada Python berjalan *preprocessing* citra, dengan hasil berikut:



Gambar 4. 6. Citra asli dari kamera smartphone

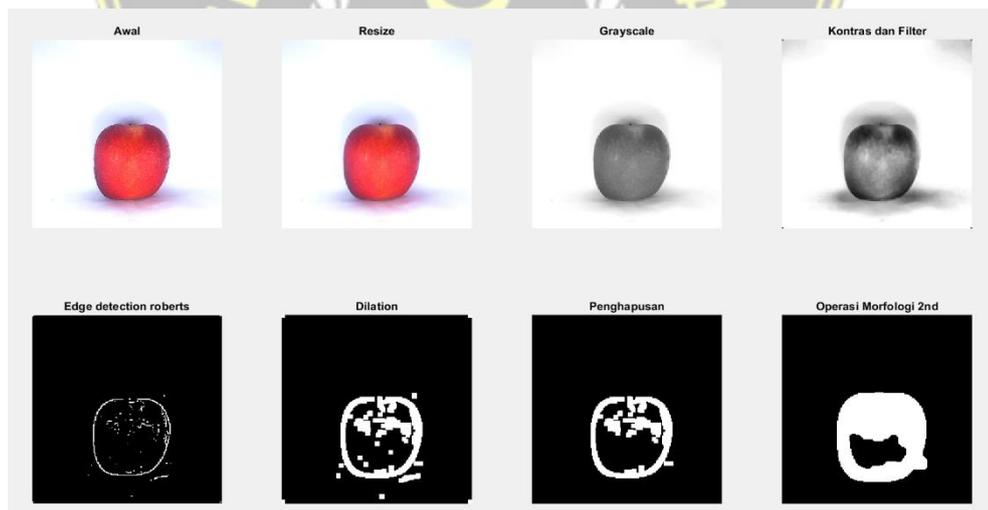


Gambar 4. 7. Hasil *resize preprocessing*



Gambar 4. 8. Hasil citra akhir *preprocessing*

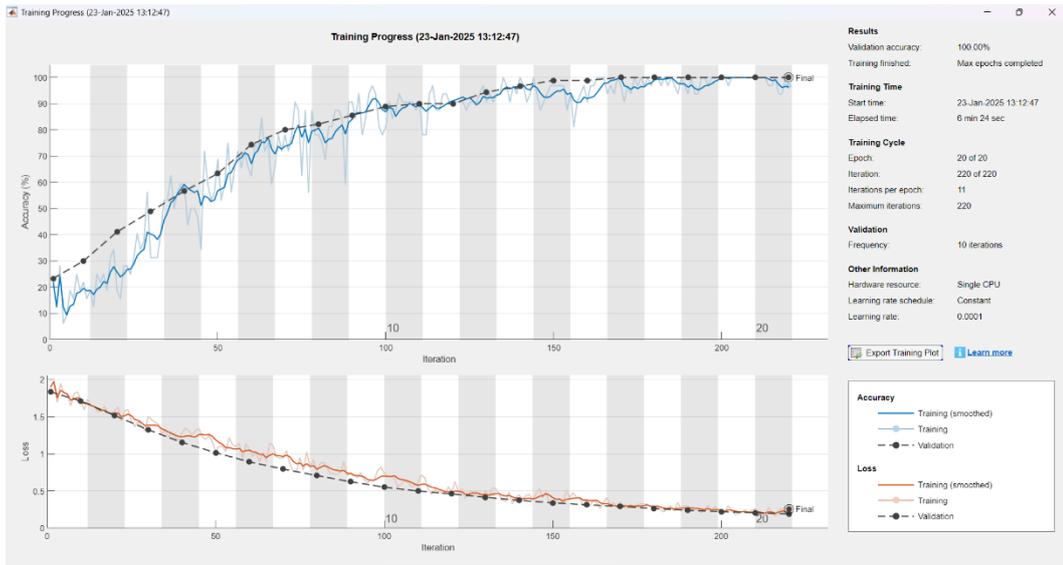
Setelah melewati *preprocessing*, hasil citra akhir menjadi *input* pada *postprocessing* pada MATLAB. Hasil tahapan *postprocessing* sebagai berikut:



Gambar 4. 9. Hasil *postprocessing* citra

4.1.4 Hasil Perancangan Training Data

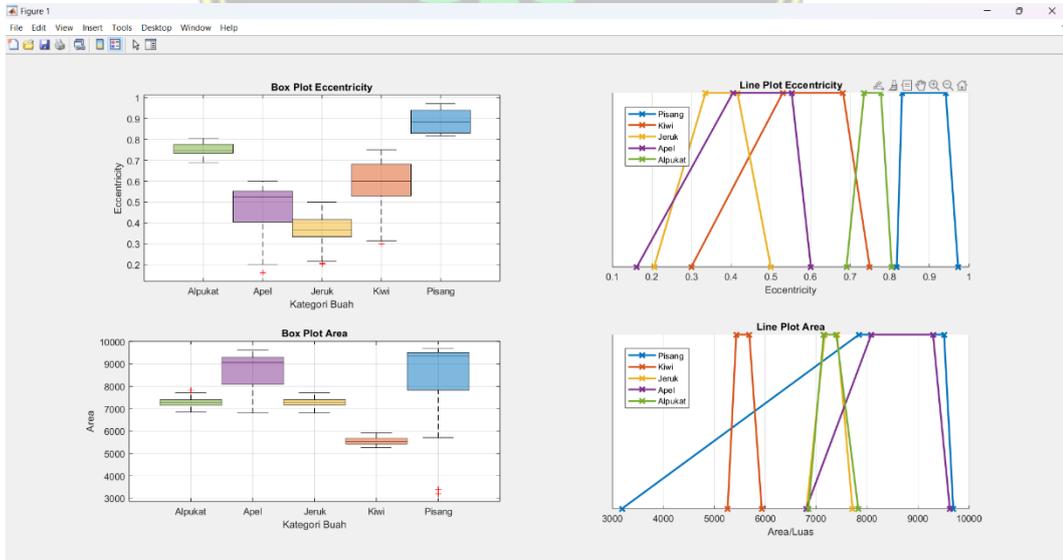
Proses *training* data terbagi menjadi *training* warna dan bentuk, maka hasil *training* juga akan menjadi dua bagian. Berikut ini adalah hasil dari kedua *training* tersebut:



Gambar 4. 10. Grafik *training progress* CNN untuk deteksi warna

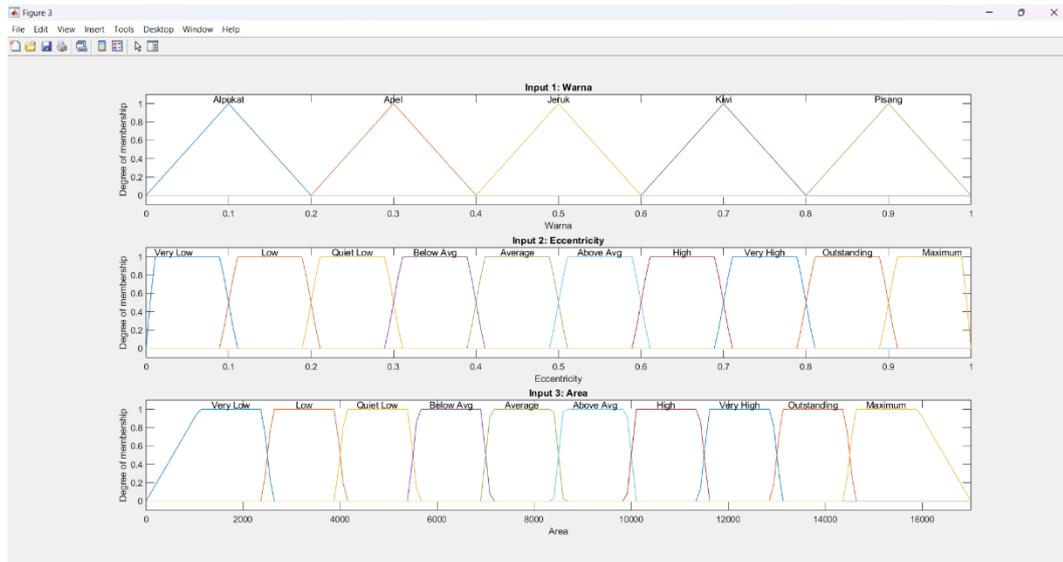
1	2	3	4	5	6	7	8	9	10	11	12	13	14
Fruit	Min_Eccentricity	Q1_Eccentricity	Median_Eccentricity	Q3_Eccentricity	Max_Eccentricity	Avg_Eccentricity	Min_Area	Q1_Area	Median_Area	Q3_Area	Max_Area	Avg_Area	
1 'Alpukat'	0.6908	0.7353	0.7474	0.7777	0.8050	0.7521	6847	7147	7.2695e+03	7395	7826	7.2749e+03	
2 'Apel'	0.1616	0.4041	0.5249	0.5524	0.6009	0.4794	6802	8082	9.0785e+03	9294	9624	8.7403e+03	
3 'Jeruk'	0.2050	0.3347	0.3660	0.4155	0.4996	0.3706	6810	7168	7.2715e+03	7410	7713	7.2779e+03	
4 'Kiwi'	0.2990	0.5299	0.6007	0.6822	0.7494	0.5933	5261	5430	5.5235e+03	5686	5930	5.5508e+03	
5 'Pisang'	0.8178	0.8309	0.8848	0.9413	0.9726	0.8965	3186	7838	9367	9511	9692	8.0752e+03	

Gambar 4. 11. Hasil perhitungan nilai statistik citra



Gambar 4. 12. Box plot dan grafik hasil perhitungan nilai statistik citra

Dari nilai statistik yang diperoleh berdasarkan *database* maka diperoleh bahwa rentang nilai eksentrisitas adalah 0 – 1 dan rentang nilai area adalah 0 – 17000, kemudian pada rentang tersebut dibagi menjadi 10 kategori, yaitu *Very Low*, *Low*, *Quiet Low*, *Below Average*, *Average*, *Above Average*, *High*, *Very High*, *Outstanding*, *Maximum*.



Gambar 4. 13. Kategori *input output fuzzy*

Dari nilai statistik dan kategori *input* terbuatlh *rules fuzzy* dengan kombinasi menyesuaikan nilai dari setiap kategori. Berikut ini adalah *rules fuzzy* yang terbentuk berdasarkan nilai statistik citra:

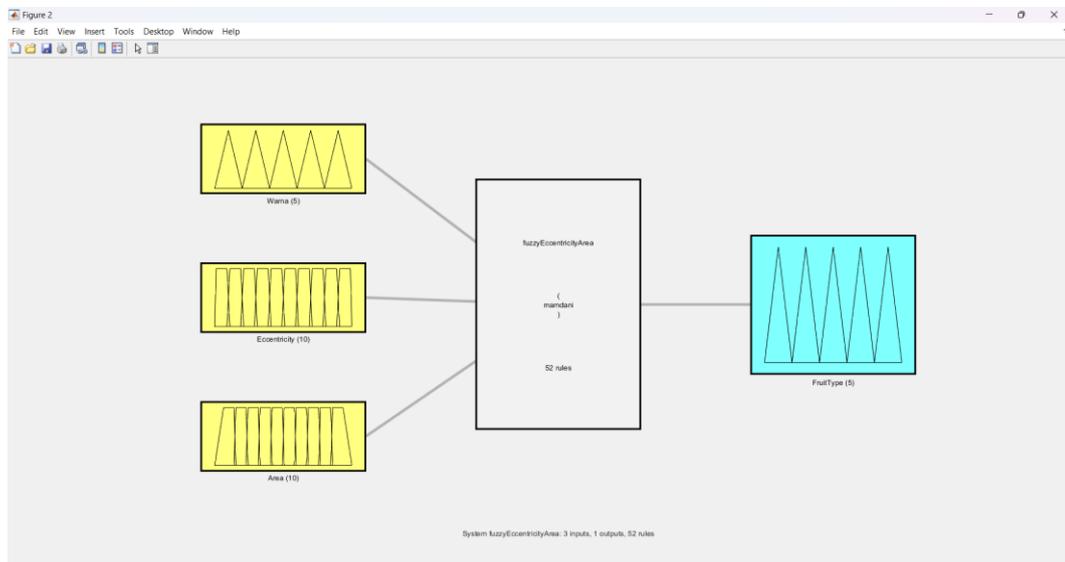
1. *If (Warna is Alpukat) and (Eccentricity is High) and (Area is Below Avg) then (FruitType is Alpukat) (0.4)*
2. *If (Warna is Alpukat) and (Eccentricity is High) and (Area is Average) then (FruitType is Alpukat) (0.5)*
3. *If (Warna is Alpukat) and (Eccentricity is Very High) and (Area is Below Avg) then (FruitType is Alpukat) (0.7)*
4. *If (Warna is Alpukat) and (Eccentricity is Very High) and (Area is Average) then (FruitType is Alpukat) (0.9)*
5. *If (Warna is Alpukat) and (Eccentricity is Outstanding) and (Area is Below Avg) then (FruitType is Alpukat) (0.4)*

6. *If (Warna is Alpukat) and (Eccentricity is Outstanding) and (Area is Average) then (FruitType is Alpukat) (0.5)*
7. *If (Warna is Apel) and (Eccentricity is Low) and (Area is Below Avg) then (FruitType is Apel) (0.5)*
8. *If (Warna is Apel) and (Eccentricity is Low) and (Area is Average) then (FruitType is Apel) (0.8)*
9. *If (Warna is Apel) and (Eccentricity is Low) and (Area is Above Avg) then (FruitType is Apel) (0.7)*
10. *If (Warna is Apel) and (Eccentricity is Quiet Low) and (Area is Below Avg) then (FruitType is Apel) (0.7)*
11. *If (Warna is Apel) and (Eccentricity is Quiet Low) and (Area is Average) then (FruitType is Apel) (1.0)*
12. *If (Warna is Apel) and (Eccentricity is Quiet Low) and (Area is Above Avg) then (FruitType is Apel) (0.9)*
13. *If (Warna is Apel) and (Eccentricity is Below Avg) and (Area is Below Avg) then (FruitType is Apel) (0.7)*
14. *If (Warna is Apel) and (Eccentricity is Below Avg) and (Area is Average) then (FruitType is Apel) (1.0)*
15. *If (Warna is Apel) and (Eccentricity is Below Avg) and (Area is Above Avg) then (FruitType is Apel) (0.9)*
16. *If (Warna is Apel) and (Eccentricity is Average) and (Area is Below Avg) then (FruitType is Apel) (0.7)*
17. *If (Warna is Apel) and (Eccentricity is Average) and (Area is Average) then (FruitType is Apel) (1.0)*
18. *If (Warna is Apel) and (Eccentricity is Average) and (Area is Above Avg) then (FruitType is Apel) (0.9)*
19. *If (Warna is Apel) and (Eccentricity is Above Avg) and (Area is Below Avg) then (FruitType is Apel) (0.7)*
20. *If (Warna is Apel) and (Eccentricity is Above Avg) and (Area is Average) then (FruitType is Apel) (1.0)*

21. *If (Warna is Apel) and (Eccentricity is Above Avg) and (Area is Above Avg) then (FruitType is Apel) (0.9)*
22. *If (Warna is Apel) and (Eccentricity is High) and (Area is Below Avg) then (FruitType is Apel) (0.4)*
23. *If (Warna is Apel) and (Eccentricity is High) and (Area is Average) then (FruitType is Apel) (0.7)*
24. *If (Warna is Apel) and (Eccentricity is High) and (Area is Above Avg) then (FruitType is Apel) (0.6)*
25. *If (Warna is Jeruk) and (Eccentricity is Quiet Low) and (Area is Below Avg) then (FruitType is Jeruk) (0.7)*
26. *If (Warna is Jeruk) and (Eccentricity is Quiet Low) and (Area is Average) then (FruitType is Jeruk) (0.8)*
27. *If (Warna is Jeruk) and (Eccentricity is Below Avg) and (Area is Below Avg) then (FruitType is Jeruk) (0.7)*
28. *If (Warna is Jeruk) and (Eccentricity is Below Avg) and (Area is Average) then (FruitType is Jeruk) (0.8)*
29. *If (Warna is Jeruk) and (Eccentricity is Average) and (Area is Below Avg) then (FruitType is Jeruk) (0.7)*
30. *If (Warna is Jeruk) and (Eccentricity is Average) and (Area is Average) then (FruitType is Jeruk) (0.8)*
31. *If (Warna is Kiwi) and (Eccentricity is Quiet Low) and (Area is Quiet Low) then (FruitType is Kiwi) (0.4)*
32. *If (Warna is Kiwi) and (Eccentricity is Quiet Low) and (Area is Below Avg) then (FruitType is Kiwi) (0.4)*
33. *If (Warna is Kiwi) and (Eccentricity is Below Avg) and (Area is Quiet Low) then (FruitType is Kiwi) (0.7)*
34. *If (Warna is Kiwi) and (Eccentricity is Below Avg) and (Area is Below Avg) then (FruitType is Kiwi) (0.8)*
35. *If (Warna is Kiwi) and (Eccentricity is Average) and (Area is Quiet Low) then (FruitType is Kiwi) (0.7)*

36. *If (Warna is Kiwi) and (Eccentricity is Average) and (Area is Below Avg) then (FruitType is Kiwi) (0.8)*
37. *If (Warna is Kiwi) and (Eccentricity is Above Avg) and (Area is Quiet Low) then (FruitType is Kiwi) (0.7)*
38. *If (Warna is Kiwi) and (Eccentricity is Above Avg) and (Area is Below Avg) then (FruitType is Kiwi) (0.8)*
39. *If (Warna is Kiwi) and (Eccentricity is High) and (Area is Quiet Low) then (FruitType is Kiwi) (0.7)*
40. *If (Warna is Kiwi) and (Eccentricity is High) and (Area is Below Avg) then (FruitType is Kiwi) (0.8)*
41. *If (Warna is Kiwi) and (Eccentricity is Very High) and (Area is Quiet Low) then (FruitType is Kiwi) (0.6)*
42. *If (Warna is Kiwi) and (Eccentricity is Very High) and (Area is Below Avg) then (FruitType is Kiwi) (0.6)*
43. *If (Warna is Pisang) and (Eccentricity is Outstanding) and (Area is Low) then (FruitType is Pisang) (0.8)*
44. *If (Warna is Pisang) and (Eccentricity is Outstanding) and (Area is Quiet Low) then (FruitType is Pisang) (0.9)*
45. *If (Warna is Pisang) and (Eccentricity is Outstanding) and (Area is Below Avg) then (FruitType is Pisang) (0.9)*
46. *If (Warna is Pisang) and (Eccentricity is Outstanding) and (Area is Average) then (FruitType is Pisang) (0.9)*
47. *If (Warna is Pisang) and (Eccentricity is Outstanding) and (Area is Above Avg) then (FruitType is Pisang) (0.9)*
48. *If (Warna is Pisang) and (Eccentricity is Maximum) and (Area is Low) then (FruitType is Pisang) (0.8)*
49. *If (Warna is Pisang) and (Eccentricity is Maximum) and (Area is Quiet Low) then (FruitType is Pisang) (0.9)*
50. *If (Warna is Pisang) and (Eccentricity is Maximum) and (Area is Below Avg) then (FruitType is Pisang) (0.9)*

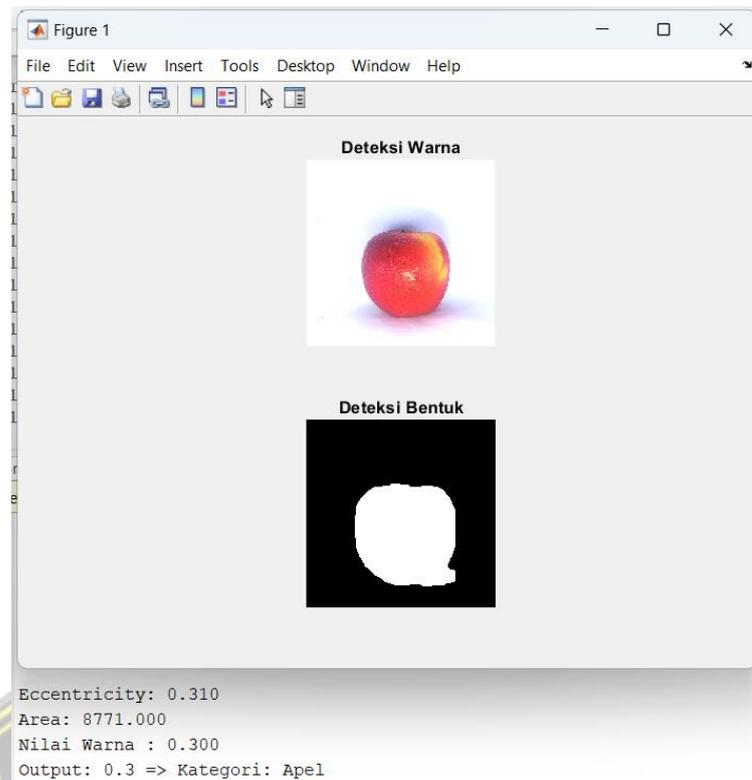
51. *If (Warna is Pisang) and (Eccentricity is Maximum) and (Area is Average) then (FruitType is Pisang) (0.9)*
52. *If (Warna is Pisang) and (Eccentricity is Maximum) and (Area is Above Avg) then (FruitType is Pisang) (0.8)*



Gambar 4. 14. Diagram blok *rules fuzzy*

4.1.5 Hasil Perancangan Proses Deteksi

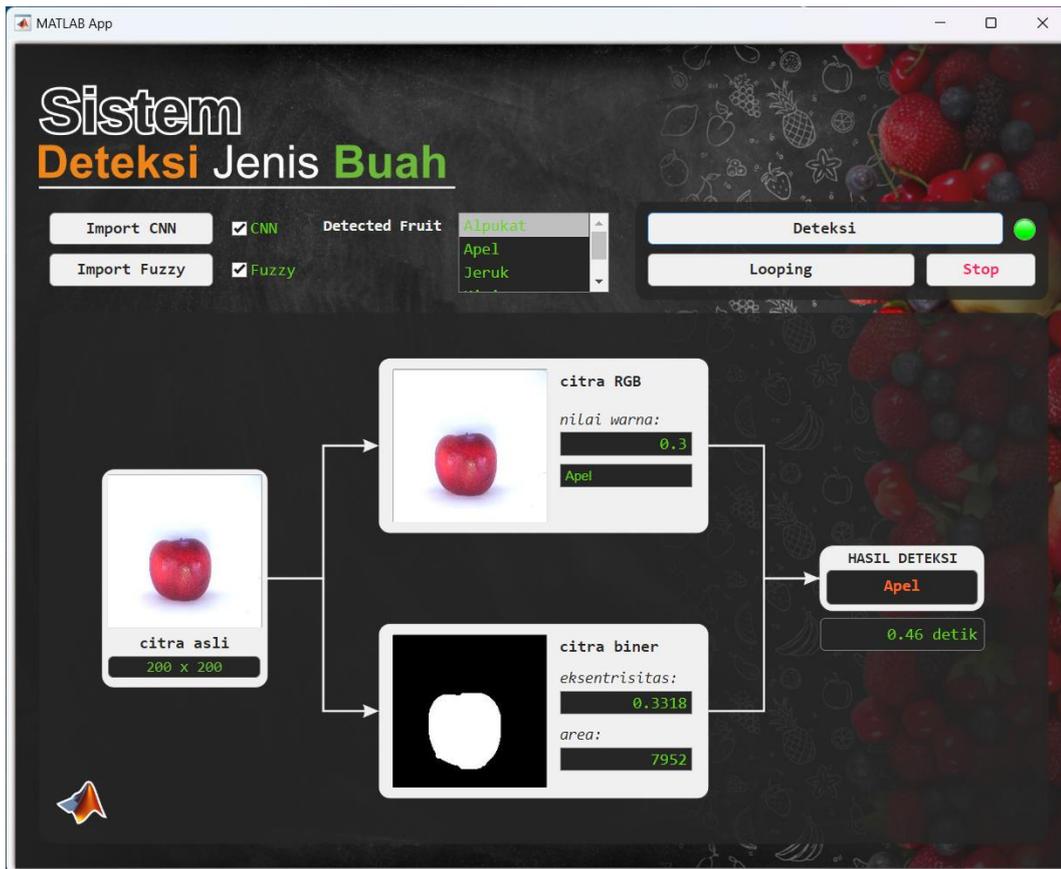
Hasil dari proses deteksi ditampilkan pada figure MATLAB dan nilainya ditampilkan pada command windows MATLAB.



Gambar 4. 15. Tampilan hasil proses deteksi

4.1.6 Hasil Perancangan Interface

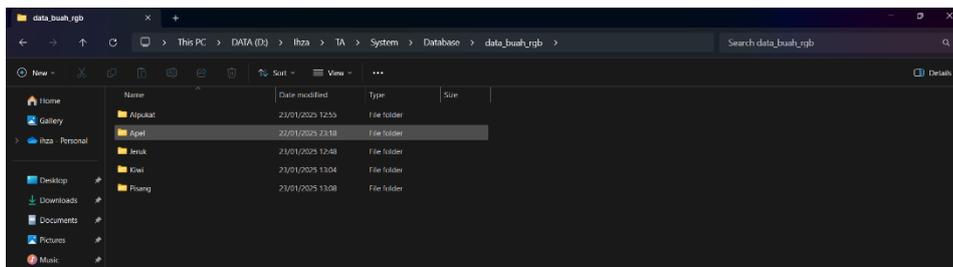
Hasil dari proses perancangan interface terdapat dua platform yang memberikan visual, yaitu *Apps Designer* MATLAB dan *live* kamera Python yang akan disandingkan untuk mempermudah proses deteksi. *Apps Designer* sesuai dengan apa yang telah dirancang. Untuk tampilan visualnya seperti pada gambar berikut ini.



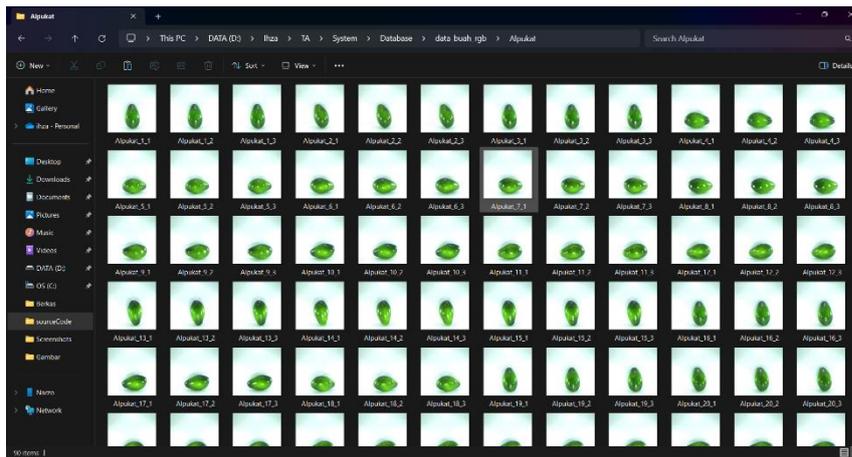
Gambar 4. 16. Interface deteksi buah (live kamera dan *Apps Designer*)

4.1.7 Hasil Perancangan *Database*

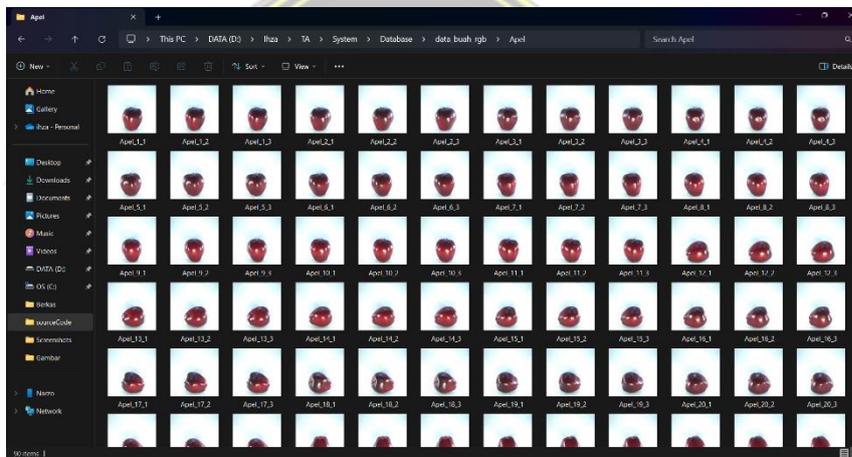
Database dirancang pada satu buah folder yang berisi subfolder-subfolder sesuai kelas (jenis buah), setiap subfolder berisi gambar-gambar buah dengan ekstensi (.jpg).



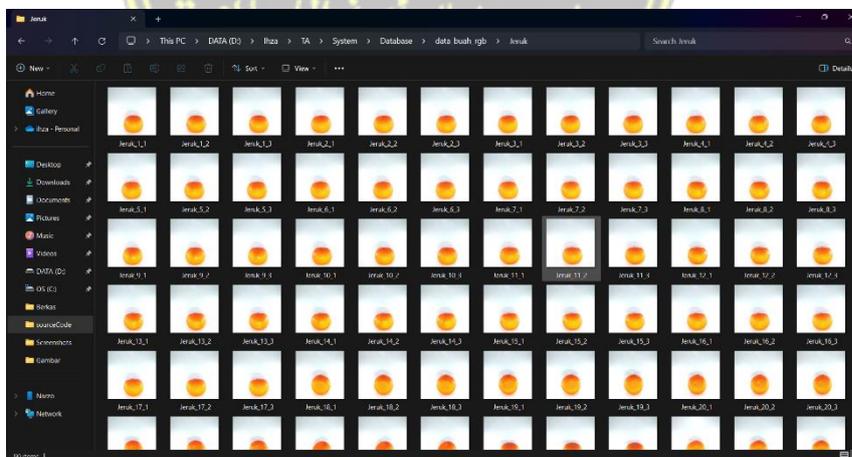
Gambar 4. 17. Hasil pengkategorian subfolder *database*



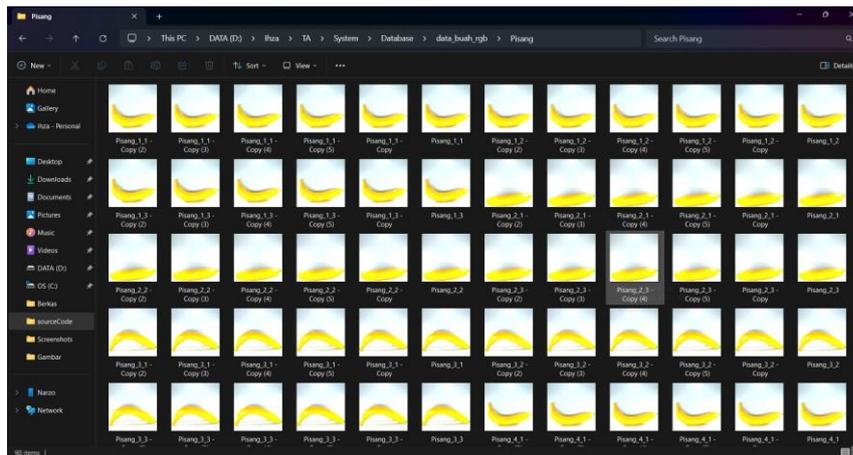
Gambar 4. 18. Gambar pada subfolder Alpakat



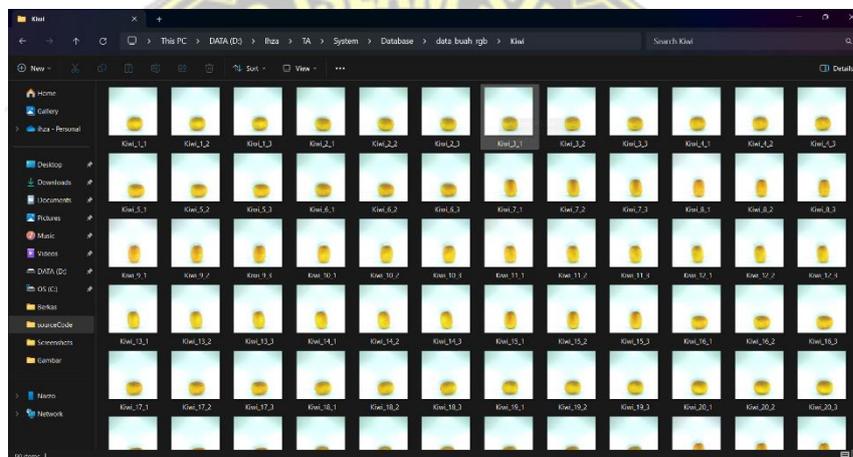
Gambar 4. 19. Gambar pada subfolder Apel



Gambar 4. 20. Gambar pada subfolder Jeruk



Gambar 4. 21. Gambar pada subfolder Pisang



Gambar 4. 22. Gambar pada subfolder Kiwi

4.2. Pengujian dan Analisis Hasil Pengujian

Pada tahap ini dilakukan pengujian yang bertujuan untuk mengetahui apakah sistem sudah terintegrasi dengan benar, sehingga sistem deteksi dapat berfungsi dengan baik. Pengujian dilakukan berdasarkan dengan rancangan pengujian yang telah dibuat. Namun untuk pengujian modul sistem akan diubah menjadi beberapa parameter, yaitu parameter model CNN, parameter processing citra, dan parameter deteksi fitur.

4.2.1 Pengujian Perangkat Keras

Pengujian perangkat keras dilakukan dengan memastikan bahwa perangkat keras bekerja dengan baik tanpa kendala.

Tabel 4. 1. Hasil pengujian perangkat keras

No	Perangkat keras	Pengujian	Hasil pengujian
1	Kamera smartphone	Menguji kamera smartphone untuk menangkap gambar dengan baik.	Baik, tanpa kendala
2	PC	Menguji PC untuk menjalankan sistem dengan baik dan lancar pada MATLAB maupun Python	Baik, tanpa kendala
3	Monitor	Menguji monitor bawaan laptop/PC untuk memberikan visual sistem dengan baik	Baik, tanpa kendala
4	Studio <i>mini</i>	Menguji studio <i>mini</i> untuk melancarkan proses deteksi dengan mencegah adanya bayangan dan memberikan pencahayaan terkontrol	Cukup, masih terdapat sedikit bayangan

4.2.2 Pengujian Platform Perangkat Lunak

Pengujian platform perangkat lunak dilakukan untuk memastikan setiap platform bekerja dengan baik sesuai kebutuhan sistem.

Tabel 4. 2. Hasil pengujian platform perangkat lunak

No	Platform	Pengujian	Hasil pengujian
1	MATLAB	Menguji MATLAB untuk melakukan proses komputasi dan image processing dengan lancar.	Baik, tanpa kendala
2	VsCode	Menguji VsCode untuk melakukan debugging program python dan menjalankannya.	Baik, tanpa kendala
3	Python	Menguji program python untuk mengakses kamera smartphone melalui Droidcam dan capture gambar serta melakukan image processing.	Baik, tanpa kendala
4	Droidcam Windows Client	Menguji Droidcam Windows Client untuk menghubungkan kamera smartphone dengan PC melalui WiFi.	Baik, tanpa kendala
5	Droidcam Classic	Menguji Droidcam Classic untuk menampilkan IP address dan port untuk koneksi ke PC.	Baik, tanpa kendala

4.2.3 Pengujian Model CNN

Pengujian model CNN bertujuan untuk mengetahui seberapa dalam CNN mempelajari sebuah *database* sehingga dapat digunakan untuk mendeteksi buah.

1. Pengujian berdasarkan resolusi citra

Pengujian ini dilakukan dengan mengubah *input size* untuk mengetahui perbedaan kecepatan pelatihan dan akurasi. Opsi pelatihan yang digunakan beragam, yaitu:

- *Database* : 450 gambar (90 gambar/kelas)

- *Learning rate* : 0,0001
- *Max Epoch* : 20
- *Mini batch size* : 32
- *Validation frequency* : 10

Tabel 4. 3. Hasil pengujian dengan perubahan *input size*

No	<i>Input Size</i>	Kecepatan	Akurasi
1	50 x 50	1 min 6 sec	58,89%
2	100 x 100	2 min 38 sec	88,89%
3	150 x 150	3 min 4 sec	98,89%
4	200 x 200	3 min 26 sec	100%

Pada table 4.1 diatas, merupakan hasil pengujian dengan melakukan perubahan *input size* dari 50 hingga 200. Dengan hasil bahwa semakin besar *input size* maka waktu pelatihan yang dibutuhkan semakin lama, namun semakin besar *input size*, tingkat akurasi yang didapatkan juga semakin tinggi. Dalam pengujian membuktikan bahwa *input size* 200x200 sudah cukup untuk digunakan dalam model CNN ini, karena akurasi yang didapat sudah mencapai 100% dan waktu yang dibutuhkan masih dibawah 5 menit.

2. Pengujian berdasarkan opsi pelatihan

Pengujian ini dilakukan dengan mengubah parameter pelatihan (*learning rate, mini batch size, max epoch*) untuk mengetahui perbedaan jumlah literasi, kecepatan pelatihan, dan akurasi. *Input* pelatihan yang digunakan seragam, yaitu:

- *Database* : 450 gambar (90 gambar/kelas)
- *Input size* : 200x200

Tabel 4. 4. Hasil pengujian dengan perubahan parameter pelatihan

No	<i>Learning rate</i>	<i>Mini batch size</i>	<i>Max epoch</i>	Jumlah iterasi	Kecepatan (sec)	Akurasi (%)
1	0,1	32	5	55	127	60
2	0,01	32	5	55	104	100
3	0,001	32	5	55	106	100
4	0,0001	32	5	55	139	67,78
5	0,00001	32	5	55	119	21,11
6	0,000001	32	5	55	91	20
7	0,1	64	5	25	119	20
8	0,01	64	5	25	106	100
9	0,001	64	5	25	107	100
10	0,0001	64	5	25	101	48,89
11	0,00001	64	5	25	159	26,67
12	0,000001	64	5	25	128	22,22
13	0,1	32	10	110	139	40
14	0,01	32	10	110	157	100
15	0,001	32	10	110	158	100
16	0,0001	32	10	110	136	93,33
17	0,00001	32	10	110	163	24,44
18	0,000001	32	10	110	144	15,56
19	0,1	64	10	50	145	100
20	0,01	64	10	50	219	100
21	0,001	64	10	50	371	100
22	0,0001	64	10	50	335	67,78
23	0,00001	64	10	50	348	36,67
24	0,000001	64	10	50	298	27,78
25	0,1	32	20	220	255	20
26	0,01	32	20	220	292	100

Tabel 4. 5. Hasil pengujian dengan perubahan parameter pelatihan

27	0,001	32	20	220	293	100
28	0,0001	32	20	220	384	100
29	0,00001	32	20	220	417	46,67
30	0,000001	32	20	220	270	6,67
31	0,1	64	20	100	235	60
32	0,01	64	20	100	243	100
33	0,001	64	20	100	210	100
34	0,0001	64	20	100	238	85,56
35	0,00001	64	20	100	273	55,56
36	0,000001	64	20	100	275	15,56

Berdasarkan tabel hasil pengujian berdasarkan perubahan parameter pelatihan, analisis dari pengaruh parameter *learning rate*, *mini-batch size*, dan *max epoch* terhadap performa model:

- a. Pengaruh *learning rate*, *learning rate* besar (0,1) menghasilkan kecepatan pelatihan yang relatif cepat, tetapi akurasi cenderung tidak stabil dan rendah pada sebagian besar kombinasi parameter. Contohnya, pada *mini-batch size* 32 dan *max epoch* 10, akurasi hanya mencapai 40%, dan pada *max epoch* 20, akurasi turun menjadi 20%. Hal ini menunjukkan bahwa *learning rate* besar dapat menyebabkan pelatihan tidak stabil dan kemungkinan overfitting. Sebaliknya, *learning rate* sedang (0,01 - 0,001) menunjukkan hasil terbaik dengan akurasi konsisten mencapai 100% pada berbagai kombinasi parameter, baik dengan *mini-batch size* 32 maupun 64, dan *max epoch* 5, 10, atau 20. Sebagai contoh, *learning rate* 0,01 dengan *mini-batch size* 32 dan *max epoch* 10 menghasilkan akurasi 100% dengan waktu pelatihan 157 detik. *Learning rate* kecil (0,0001 - 0,000001) meningkatkan waktu komputasi secara signifikan tanpa memberikan keuntungan berarti dalam akurasi. Pada *learning rate* 0,000001 dengan *mini-batch size* 32 dan *max epoch* 10, akurasi hanya mencapai 15,56%.

Namun, *learning rate* 0,0001 dapat menghasilkan akurasi 100% pada *max epoch* 20 dengan *mini-batch size* 32, meskipun waktu pelatihan lebih lama (384 detik). Oleh karena itu, *learning rate* optimal adalah 0,01 atau 0,001 karena memberikan akurasi tinggi dengan efisiensi waktu pelatihan yang baik.

- b. Pengaruh *mini-batch size*, *mini-batch size* 32 memiliki kecepatan pelatihan yang lebih tinggi dibandingkan *batch size* 64, terutama pada *max epoch* rendah. Sebagai contoh, pada *learning rate* 0,01 dan *max epoch* 5, waktu pelatihan hanya 104 detik dengan akurasi mencapai 100%. Namun, untuk *max epoch* yang lebih tinggi, seperti 20, waktu pelatihan meningkat menjadi 292 detik, tetapi akurasi tetap konsisten pada 100%. Di sisi lain, *mini-batch size* 64 memiliki waktu pelatihan yang lebih lambat tetapi memberikan akurasi tinggi pada konfigurasi tertentu. Sebagai contoh, pada *learning rate* 0,001 dan *max epoch* 10, akurasi mencapai 100% dengan waktu pelatihan 371 detik, dan pada *learning rate* 0,01 dengan *max epoch* 20, waktu pelatihan adalah 243 detik dengan akurasi tetap 100%. Dengan demikian, *mini-batch size* 32 lebih efisien untuk waktu pelatihan, sementara *batch size* 64 lebih cocok untuk *database* besar atau *max epoch* yang tinggi.
- c. Pengaruh *max epoch*, *max epoch* rendah (5) menghasilkan waktu pelatihan yang lebih cepat, tetapi akurasi sangat bergantung pada *learning rate*. Sebagai contoh, dengan *learning rate* 0,01 dan *mini-batch size* 64, akurasi mencapai 100% dalam waktu 106 detik, sedangkan pada *learning rate* 0,00001 dengan *mini-batch size* 32, akurasi hanya 21,11%. *Max epoch* sedang (10) menunjukkan akurasi yang lebih konsisten dengan *learning rate* optimal (0,01 - 0,001). Misalnya, pada *learning rate* 0,01 dan *mini-batch size* 32, akurasi tetap 100% dengan waktu pelatihan 157 detik. Pada *max epoch* tinggi (20), akurasi cenderung stabil pada berbagai nilai *learning rate*. Sebagai contoh, *learning rate* 0,0001 dengan *mini-batch size* 32 memberikan akurasi 100% dengan waktu pelatihan 384 detik. Namun, *max epoch* yang tinggi juga membutuhkan waktu pelatihan yang

lebih lama. Dengan demikian, *max epoch* 20 memberikan akurasi terbaik tetapi memerlukan waktu pelatihan yang lebih signifikan.

Parameter terbaik berdasarkan tabel, parameter terbaik untuk model adalah *learning rate* 0,01, *mini-batch size* 32, dan *max epoch* 20. Kombinasi ini memberikan akurasi 100% dengan waktu pelatihan 292 detik, menjadikannya konfigurasi yang optimal untuk efisiensi dan performa sistem. Namun untuk sistem deteksi yang lebih akurat terhadap detail kecil pada gambar maka *learning rate* 0,0001, *mini-batch size* 32, dan *max epoch* 20 adalah kombinasi yang terbaik walaupun mengorbankan waktu pelatihan yang lebih lama.

Dari keseluruhan pengujian model CNN didapat parameter terbaik untuk menjalankan sistem adalah dengan *input size* 200x200, *learning rate* 0,00001, *mini-batch size* 32, dan *max epoch* 20.

4.2.4 Pengujian Processing Citra

Pengujian processing citra diuji dengan melihat pengaruh latar belakang dan cahaya terkontrol terhadap processing citra.

Tabel 4. 6. Hasil pengujian berdasarkan latar belakang dan pencahayaan

No	Latar belakang	Pencahayaan	Hasil
1	Putih	Menyala	Sangat baik, tanpa noise
2	Putih	Mati	Cukup baik, terdapat noise
3	Hitam	Menyala	Kurang baik, banyak noise
4	Hitam	Mati	Sangat kurang baik, noise mendominasi
5	Merah	Menyala	Baik, sedikit noise
6	Merah	Mati	Sangat kurang baik, noise mendominasi
7	Biru	Menyala	Kurang baik, banyak noise
8	Biru	Mati	Sangat kurang baik, noise mendominasi
9	Hijau	Menyala	Baik, sedikit noise
10	Hijau	Mati	Sangat kurang baik, noise mendominasi

Dari pengujian, kualitas gambar sangat dipengaruhi oleh latar belakang dan kondisi pencahayaan. Pengujian menunjukkan bahwa kualitas gambar terbaik diperoleh pada latar belakang putih dengan pencahayaan menyala, menghasilkan gambar tanpa noise. Latar putih dengan pencahayaan mati tetap cukup baik, meski terdapat sedikit noise. Latar hitam, merah, biru, dan hijau menunjukkan penurunan kualitas saat pencahayaan mati, dengan noise yang mendominasi. Latar merah dan hijau menghasilkan kualitas baik dengan sedikit noise ketika pencahayaan menyala, sementara latar biru dan hitam tetap memiliki banyak noise meskipun pencahayaan aktif. Secara keseluruhan, latar putih dengan pencahayaan menyala adalah kondisi optimal untuk gambar berkualitas tinggi. Latar belakang merah, hijau, dan biru juga akan memperburuk hasil jika buah yang dideteksi berwarna serupa.

4.2.5 Pengujian Sistem Deteksi

Pengujian ini difungsikan untuk mengetahui akurasi sistem dan mengetahui seberapa lama waktu yang dihabiskan oleh sistem dari akuisisi citra sampai menampilkan hasil deteksi. Pengujian ini terbagi menjadi tiap jenis buah yang dilakukan 10 kali pengujian tiap buahnya. Model CNN yang digunakan pada pengujian ini menggunakan:

- *Database* : 450 gambar (90 gambar/kelas)
- *Latar belakang* : putih
- *Pencahayaan* : menyala
- *Learning rate* : 0,00001
- *Mini-batch size* : 32
- *Max epoch* : 20

Tabel 4. 7. Hasil pengujian sistem deteksi pada alpukat

No Pengujian	Kecepatan deteksi (detik)	Buah terdeteksi	Keterangan
Pengujian ke-1	0,22	Alpukat	Benar
Pengujian ke-2	0,23	Alpukat	Benar
Pengujian ke-3	0,23	Alpukat	Benar

Tabel 4. 8. Hasil pengujian sistem deteksi pada alpukat

Pengujian ke-4	0,22	Alpukat	Benar
Pengujian ke-5	0,21	Alpukat	Benar
Pengujian ke-6	0,23	Alpukat	Benar
Pengujian ke-7	0,22	Alpukat	Benar
Pengujian ke-8	0,21	Alpukat	Benar
Pengujian ke-9	0,24	Alpukat	Benar
Pengujian ke-10	0,21	Alpukat	Benar

Pengujian deteksi alpukat menunjukkan hasil yang sangat baik dengan akurasi 100%. Semua pengujian berhasil mendeteksi alpukat dengan benar. Rata-rata waktu deteksi adalah 0,222 detik, dengan waktu tercepat 0,21 detik dan waktu terlambat 0,24 detik. Konsistensi sistem dalam mendeteksi alpukat menunjukkan keandalan model untuk kelas buah ini, tanpa adanya kesalahan atau anomali selama pengujian.

Tabel 4. 9. Hasil pengujian sistem deteksi pada apel

No Pengujian	Kecepatan deteksi (detik)	Buah terdeteksi	Keterangan
Pengujian ke-1	0,25	Apel	Benar
Pengujian ke-2	0,25	Apel	Benar
Pengujian ke-3	0,24	Apel	Benar
Pengujian ke-4	0,24	Apel	Benar
Pengujian ke-5	0,27	Apel	Benar
Pengujian ke-6	0,23	Apel	Benar
Pengujian ke-7	0,23	Apel	Benar
Pengujian ke-8	0,22	Jeruk	Salah
Pengujian ke-9	0,24	Apel	Benar
Pengujian ke-10	0,24	Apel	Benar

Pada pengujian deteksi apel, sistem mencapai akurasi 90%, dengan 9 dari 10 pengujian berhasil mendeteksi apel dengan benar. Satu kesalahan terjadi pada pengujian ke-8, di mana apel terdeteksi sebagai jeruk. Waktu deteksi rata-rata adalah 0,241 detik, dengan waktu tercepat 0,22 detik dan waktu terlambat 0,27 detik. Kesalahan ini menunjukkan adanya potensi untuk perbaikan dalam membedakan apel dari kelas buah lain, meskipun secara keseluruhan performa sistem cukup baik.

Tabel 4. 10. Hasil pengujian sistem deteksi pada jeruk

No Pengujian	Kecepatan deteksi (detik)	Buah terdeteksi	Keterangan
Pengujian ke-1	0,27	Jeruk	Benar
Pengujian ke-2	0,25	Jeruk	Benar
Pengujian ke-3	0,24	Jeruk	Benar
Pengujian ke-4	0,23	Jeruk	Benar
Pengujian ke-5	0,23	Jeruk	Benar
Pengujian ke-6	0,23	Jeruk	Benar
Pengujian ke-7	0,23	Jeruk	Benar
Pengujian ke-8	0,36	Jeruk	Benar
Pengujian ke-9	0,22	Jeruk	Benar
Pengujian ke-10	0,23	Jeruk	Benar

Sistem menunjukkan performa yang sangat baik dalam mendeteksi jeruk, dengan akurasi 100% di semua pengujian. Waktu deteksi rata-rata adalah 0,249 detik, dengan waktu tercepat 0,22 detik dan waktu terlambat 0,36 detik. Meskipun terdapat sedikit peningkatan waktu deteksi pada pengujian ke-8, sistem tetap konsisten dalam mendeteksi jeruk tanpa kesalahan klasifikasi.

Tabel 4. 11. Hasil pengujian sistem deteksi pada kiwi

No Pengujian	Kecepatan deteksi (detik)	Buah terdeteksi	Keterangan
Pengujian ke-1	0,27	Kiwi	Benar
Pengujian ke-2	0,25	Kiwi	Benar
Pengujian ke-3	0,25	Kiwi	Benar
Pengujian ke-4	0,28	Kiwi	Benar
Pengujian ke-5	0,25	Kiwi	Benar
Pengujian ke-6	0,25	Kiwi	Benar
Pengujian ke-7	0,23	Kiwi	Benar
Pengujian ke-8	0,24	Kiwi	Benar
Pengujian ke-9	0,24	Kiwi	Benar
Pengujian ke-10	0,23	Kiwi	Benar

Pada pengujian deteksi kiwi, sistem berhasil mendeteksi dengan akurasi 100% di seluruh pengujian. Waktu deteksi rata-rata adalah 0,249 detik, dengan waktu tercepat 0,23 detik dan waktu terlambat 0,28 detik. Kinerja ini menunjukkan bahwa sistem sangat andal dan konsisten dalam mendeteksi buah kiwi tanpa adanya kesalahan klasifikasi.

Tabel 4. 12. Hasil pengujian sistem deteksi pada pisang

No Pengujian	Kecepatan deteksi (detik)	Buah terdeteksi	Keterangan
Pengujian ke-1	0,23	Pisang	Benar
Pengujian ke-2	0,24	Pisang	Benar
Pengujian ke-3	0,24	Pisang	Benar
Pengujian ke-4	0,27	Pisang	Benar
Pengujian ke-5	0,22	Pisang	Benar
Pengujian ke-6	0,23	Pisang	Benar
Pengujian ke-7	0,27	Pisang	Benar
Pengujian ke-8	0,25	Pisang	Benar

Tabel 4. 13. Hasil pengujian sistem deteksi pada pisang

Pengujian ke-9	0,25	Pisang	Benar
Pengujian ke-10	0,23	Pisang	Benar

Hasil pengujian deteksi pisang menunjukkan akurasi 100%, dengan semua pengujian mendeteksi buah pisang dengan benar. Waktu deteksi rata-rata adalah 0,244 detik, dengan waktu tercepat 0,22 detik dan waktu terlambat 0,27 detik. Performa sistem yang stabil dan cepat ini menunjukkan efektivitas model dalam mengenali buah pisang dengan baik dan konsisten.

4.2.6 Pengujian Database

Pengujian dilakukan dengan merubah jumlah gambar pada *database*, yaitu sebanyak 30 gambar/kelas (150 gambar), 60 gambar perkelas (300 gambar), dan 90 gambar/kelas (450 gambar). Untuk mengetahui pengaruh banyaknya jumlah gambar pada *database* terhadap akurasi deteksi.

Tabel 4. 14. Hasil pengujian dengan perubahan jumlah *database*

No	Jumlah <i>database</i>	Akurasi CNN	Buah sebenarnya	Buah terdeteksi	Kecepatan deteksi (detik)	Ket
1	150	43,33%	Alpukat	Alpukat	0,83	Benar
2	150	43,33%	Apel	Jeruk	0,22	Salah
3	150	43,33%	Jeruk	Jeruk	0,20	Benar
4	150	43,33%	Kiwi	Jeruk	0,26	Salah
5	150	43,33%	Pisang	Jeruk	0,19	Salah
6	300	35%	Alpukat	Jeruk	0,19	Salah
7	300	35%	Apel	Apel	0,19	Benar
8	300	35%	Jeruk	Jeruk	0,24	Benar
9	300	35%	Kiwi	Kiwi	0,20	Benar
10	300	35%	Pisang	Jeruk	0,19	Salah
11	450	100%	Alpukat	Alpukat	0,22	Benar

Tabel 4. 15. Hasil pengujian dengan perubahan jumlah *database*

12	450	100%	Apel	Apel	0,24	Benar
13	450	100%	Jeruk	Jeruk	0,24	Benar
14	450	100%	Kiwi	Kiwi	0,25	Benar
15	450	100%	Pisang	Pisang	0,23	Benar

Pengujian dilakukan dengan variasi jumlah gambar dalam *database* untuk mengetahui pengaruhnya terhadap akurasi deteksi sistem CNN. *Database* diuji dengan 30 gambar per kelas (150 gambar), 60 gambar per kelas (300 gambar), dan 90 gambar per kelas (450 gambar). Hasil menunjukkan bahwa jumlah gambar dalam *database* sangat memengaruhi akurasi deteksi. Pada *database* dengan 150 gambar, akurasi hanya mencapai 43,33%, dengan sebagian besar buah terdeteksi salah. Pada *database* ini, alpukat dan jeruk terdeteksi benar, namun apel, kiwi, dan pisang mengalami kesalahan deteksi, dengan kecepatan deteksi berkisar antara 0,19 hingga 0,83 detik.

Ketika jumlah *database* meningkat menjadi 300 gambar, akurasi menurun menjadi 35%. Walaupun apel, jeruk, dan kiwi terdeteksi dengan benar, alpukat dan pisang masih sering terdeteksi salah. Kecepatan deteksi sedikit meningkat, dengan rata-rata waktu deteksi sekitar 0,19–0,24 detik. Hal ini menunjukkan bahwa meskipun jumlah data meningkat, kualitas data atau proses pelatihan model belum optimal.

Pada *database* dengan 450 gambar, sistem mencapai akurasi 100%, di mana semua buah berhasil terdeteksi dengan benar. Kecepatan deteksi tetap konsisten di kisaran 0,22–0,25 detik. Hasil ini menunjukkan bahwa penggunaan *database* yang lebih besar dan bervariasi meningkatkan kemampuan sistem dalam mengenali pola dan fitur pada citra buah, sehingga menghasilkan deteksi yang akurat.

Secara keseluruhan, pengujian ini menunjukkan bahwa jumlah gambar pada *database* sangat berpengaruh terhadap akurasi dan stabilitas deteksi. Semakin banyak data yang digunakan, terutama dengan variasi yang mencakup berbagai

kondisi pencahayaan dan sudut pandang, semakin baik performa model CNN. Dengan akurasi 100% pada 450 gambar, sistem menunjukkan bahwa *database* yang besar dan berkualitas tinggi merupakan faktor kunci dalam mendukung deteksi yang optimal.



BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Pada pembuatan tugas akhir yang berjudul “Implementasi *Adaptive Neuro-Fuzzy Inference System* (ANFIS) Untuk Deteksi Otomatis Jenis Buah Berdasarkan Citra Warna dan Bentuk Menggunakan MATLAB”, dapat disimpulkan:

1. Metode ANFIS terbukti mampu mengintegrasikan fitur warna dan bentuk secara efektif untuk meningkatkan performa klasifikasi buah. Dengan memanfaatkan aturan *fuzzy* untuk mengintegrasikan antara *input* warna dan geometri objek. Dan kemampuan pembelajaran dari jaringan saraf tiruan, sistem dapat menyesuaikan bobot keanggotaan fitur warna untuk menghasilkan keputusan klasifikasi.
2. Kombinasi CNN untuk ekstraksi fitur warna dan Edge Detection untuk analisis bentuk telah menunjukkan efektivitas yang tinggi dalam mendeteksi karakteristik visual buah. CNN mampu menangkap perbedaan warna secara detail dengan parameter pelatihan pada *learning rate* 0,0001, *mini batch size* 32 dan 20 *epoch*. Sementara metode Edge Detection membantu dalam identifikasi bentuk melalui analisis geometri objek. Integrasi kedua metode ini meningkatkan kemampuan sistem dalam membedakan berbagai jenis buah.
3. Sistem yang dikembangkan menghasilkan tingkat akurasi deteksi yang tinggi, dengan pencapaian sebesar 99,8% berdasarkan pengujian yang telah dilakukan sebanyak 10 kali pada tiap jenis buah. Hasil ini menunjukkan bahwa pendekatan yang diterapkan memiliki keandalan yang baik dalam mengklasifikasikan buah berdasarkan warna dan bentuk dalam kondisi pencahayaan yang terkendali.

4. Penggunaan MATLAB sebagai platform simulasi terbukti sangat efektif dalam mengimplementasikan sistem deteksi buah. MATLAB menyediakan berbagai fungsi dan *toolbox* yang mendukung pemrosesan citra, pelatihan model CNN, serta penerapan logika *fuzzy* melalui ANFIS, sehingga mempercepat proses pengembangan dan validasi model yang diusulkan.
5. Hardware yang digunakan, termasuk *mini studio* dengan latar belakang putih dan pencahayaan terkendali, kamera *smartphone*, dan laptop dengan spesifikasi yang mendukung, terbukti meningkatkan akurasi deteksi. Kombinasi perangkat keras ini berhasil meminimalkan noise, bayangan, serta kesalahan deteksi, mendukung performa sistem dalam berbagai pengujian.

5.2. Saran

Setelah pembuatan tugas akhir yang berjudul “Implementasi *Adaptive Neuro-Fuzzy Inference System* (ANFIS) Untuk Deteksi Otomatis Jenis Buah Berdasarkan Citra Warna dan Bentuk Menggunakan MATLAB”, terdapat beberapa saran untuk inovasi kedepannya:

1. Melakukan eksperimen lebih lanjut pada hyperparameter CNN seperti *learning rate*, jumlah *epoch*, dan ukuran *batch* untuk meningkatkan efisiensi waktu dan akurasi deteksi.
2. Menambahkan lebih banyak jenis buah untuk memastikan kemampuan generalisasi sistem terhadap *database* yang lebih kompleks.
3. Meningkatkan performa sistem agar mampu bekerja secara real-time, terutama dengan optimasi kode dan hardware yang mendukung.
4. Menggunakan teknik pemrosesan lanjutan untuk mengurangi noise lebih efektif, misalnya melalui filter adaptif atau algoritma denoising lainnya.
5. Menguji sistem di luar ruangan pada kondisi pencahayaan alami untuk mengevaluasi kinerjanya di dunia nyata dan mengidentifikasi kebutuhan perbaikan.
6. Menggunakan hardware yang lebih mendukung pemrosesan citra pada MATLAB seperti GPU dari NVIDIA yang mendukung CUDA untuk

meningkatkan model CNN menjadi R-CNN, Fast R-CNN, bahkan Faster R-CNN.



DAFTAR PUSTAKA

- [1] Y. Yohannes, M. R. Pribadi, and L. Chandra, "Klasifikasi Jenis Buah dan Sayuran Menggunakan SVM Dengan Fitur Saliency-HOG dan Color Moments," *ELKHA: Jurnal Teknik Elektro*, vol. 12, no. 2, pp. 125–131, 2020.
- [2] I. Amal, M. Muhammad, and A. B. Kaswar, "Sistem Pendeteksi Kematangan Buah Tomat Berbasis Pengolahan Citra Digital Menggunakan Metode Jaringan Syaraf Tiruan," *Jurnal MediaTIK*, pp. 65–69, 2022.
- [3] M. Meiriyama, "Klasifikasi Citra Buah berbasis fitur warna HSV dengan klasifikator SVM," *Jurnal Komputer Terapan*, vol. 4, no. 1, pp. 50–61, 2018.
- [4] L. Agustina, D. Mahmudah, A. B. Setiawan, R. Mustika, A. Dunan, and A. Ratnawati, "Perkembangan Ekonomi Digital di Indonesia: Strategi dan Sektor Potensial," *Jakarta: Puslitbang Aptika dan IKP*, 2019.
- [5] R. Manjunath, "Dimensionality reduction and classification of color features data using svm and knn," *Int J Image Process Vis Commun*, vol. 1, pp. 16–21, 2013.
- [6] D. Savakar, "Identification and classification of bulk fruits images using artificial neural networks," *International Journal of Engineering and Innovative Technology (IJEIT)*, vol. 1, no. 3, pp. 35–40, 2012.
- [7] R. Amirulah, M. M. Mokji, and Z. Ibrahim, "Starfruit color maturity classification using Cr as feature," in *2010 Sixth International Conference on Signal-Image Technology and Internet Based Systems*, IEEE, 2010, pp. 93–97.
- [8] M. Suresha, N. A. Shilpa, and B. Soumya, "Apples grading based on SVM classifier," *Int. J. Comput. Appl*, vol. 975, p. 8878, 2012.

- [9] N. B. A. Mustafa, K. Arumugam, S. K. Ahmed, and Z. A. M. Sharrif, "Classification of fruits using Probabilistic Neural Networks-Improvement using color features," in *TENCON 2011-2011 IEEE Region 10 Conference*, IEEE, 2011, pp. 264–269.
- [10] W. C. Seng and S. H. Mirisaee, "A new method for fruits recognition system," in *2009 international conference on electrical engineering and informatics*, IEEE, 2009, pp. 130–134.
- [11] H. Anam, R. Alfita, and K. A. Wibisono, "Sistem Pendeteksi Kerusakan Jalan Berbasis Kamera dengan Algoritma Edge Detection," *SinarFe7*, vol. 1, no. 1, pp. 7–12, 2018.
- [12] N. Nurdin, D. Hamdhana, and M. J. Setiawan, "Sistem Pendeteksi Pola Lafadz Allah Dan Muhammad Pada Citra Al-Qur'an Menggunakan Metode Peirce," *TECHSI-Jurnal Teknik Informatika*, vol. 9, no. 2, pp. 78–90, 2017.
- [13] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.
- [14] R. B. Tosi, H. dorothea Mbura, and Y. R. Kaesmetan, "Implementasi CNN Dalam Mengidentifikasi Kematangan Cabai Berdasarkan Warna," *Indonesian Journal of Education And Computer Science*, vol. 2, no. 1, pp. 34–42, 2024.
- [15] F. F. Maulana and N. Rochmawati, "Klasifikasi citra buah menggunakan convolutional neural network," *Journal of Informatics and Computer Science (JINACS)*, vol. 1, no. 02, pp. 104–108, 2019.
- [16] M. Brahim, K. Boukhalifa, and A. Moussaoui, "Deep learning for tomato diseases: classification and symptoms visualization," *Applied Artificial Intelligence*, vol. 31, no. 4, pp. 299–315, 2017.
- [17] L. Zhang, G. Gui, A. M. Khattak, M. Wang, W. Gao, and J. Jia, "Multi-task cascaded convolutional networks based intelligent fruit detection for designing automated robot," *IEEE Access*, vol. 7, pp. 56028–56038, 2019.

- [18] N. Syafitri, “Pengenalan Pola Untuk Deteksi Uang Koin,” in *Seminar Nasional Teknologi Informasi Komunikasi dan Industri*, 2011.
- [19] R. C. Gonzalez, *Digital image processing*. Pearson education india, 2009.
- [20] M. Harim, H. A. T. Muslimin, and ..., “Segmentasi Citra Telapak Tangan menggunakan Deteksi Tepi Prewitt, Sobel, Roberts, dan Canny,” *JIMP-Jurnal Informatika ...*, 2023, [Online]. Available: <http://ejurnal.unmerpas.ac.id/index.php/informatika/article/view/836>
- [21] M. H. Beale, M. T. Hagan, and H. B. Demuth, “Deep learning toolbox,” *R2018b User’s Guide; The MathWorks, Inc.: Natick, MA, USA*, 2018.

