

**KLASIFIKASI KUALITAS JERUK LOKAL MENGGUNAKAN METODE
*NEURAL NETWORK***

SKRIPSI

**Oleh:
JUNAEDI ABDILLAH
NIM. 19650070**



**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2024**

**KLASIFIKASI KUALITAS JERUK LOKAL MENGGUNAKAN METODE
*NEURAL NETWORK***

SKRIPSI

**Diajukan Kepada:
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang
Untuk Memenuhi Salah Satu Persyaratan Dalam
Memperoleh Gelar Sarjana Komputer (S.Kom)**

**Oleh:
JUNAEDI ABDILLAH
NIM. 19650070**

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2024**

HALAMAN PERSETUJUAN

**KLASIFIKASI KUALITAS JERUK LOKAL MENGGUNAKAN METODE
NEURAL NETWORK**

SKRIPSI

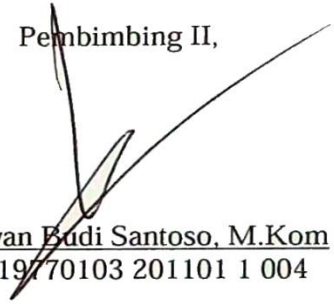
Oleh :
JUNAEDI ABDILLAH
NIM. 19650070

Telah Diperiksa dan Disetujui untuk Diuji:
Tanggal: 6 Juni 2024

Pembimbing I,



Dr. Cahyo Crysdiyan, M.Cs
NIP. 19740424 200901 1 008

Pembimbing II,


Dr. Irwan Budi Santoso, M.Kom
NIP. 19770103 201101 1 004

Mengetahui,
Ketua Program Studi Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang




Dr. Fachrul Kurniawan, M.MT, IPM
NIP. 19771020 200912 1 001

HALAMAN PENGESAHAN

**KLASIFIKASI KUALITAS JERUK LOKAL MENGGUNAKAN METODE
NEURAL NETWORK**

SKRIPSI

Oleh :
JUNAEDI ABDILLAH
NIM. 19650070

Telah Dipertahankan di Depan Dewan Penguji Skripsi
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
Untuk Memperoleh Gelar Sarjana Komputer (S.Kom)
Tanggal: 6 Juni 2024





Susunan Dewan Penguji

Ketua Penguji : Dr. Ir. Mokhamad Amin Hariyadi, M.T
NIP. 19670018 200501 1 001

Anggota Penguji I : Ajib Hanani, M.T
NIP. 19840731 202321 1 013

Anggota Penguji II : Dr. Cahyo Crysdiyan, M. Cs
NIP. 19740424 200901 1 008

Anggota Penguji III : Dr. Irwan Budi Santoso, M.Kom
NIP. 19770103 201101 1 004

()
()
()
()

Mengetahui dan Megeahkan,
Ketua Program Studi Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang



Dr. Fachrul Kurniawan, M.MT, IPM
NIP. 19771020 200912 1 001

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini:

Nama : Junaedi Abdillah

NIM : 19650070

Program Studi : Teknik Informatika

Fakultas : Sains dan Teknologi

Judul Skripsi : Klasifikasi Kualitas Jeruk Lokal Menggunakan Metode
Neural Network

Menyatakan dengan sebenarnya bahwa Skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri, bukan merupakan pengambil alihan data, tulisan, atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar pustaka.

Apabila dikemudian hari terbukti atau dapat dibuktikan skripsi ini merupakan hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 6 Juni 2024

Yang membuat pernyataan,



Junaedi Abdillah

NIM.19650070

HALAMAN MOTTO

“I have no life, just a explorer with curiosity and thirst for knowledge. This makes me feel truly alive.”

HALAMAN PERSEMBAHAN

Alhamdulillah *rabbi'l'alam*, Puji syukur kehadirat Allah SWT atas segala limpahan berkah dan rahmat dalam perjalanan menempuh ilmu studi S1 di Universitas Islam Negeri Maulana Malik Ibrahim Malang. Sholawat serta salam kepada junjungan kita Nabi Muhammad SAW yang telah membimbing kita dengan risalah kebenaran sebagai tuntunan untuk senantiasa melakukan kebajikan.

Skripsi saya persembahkan kepada penulis yang telah mencurahkan segenap tenaga dan pikiran dalam menyelesaikannya. Kepada orang tua penulis Bapak Mashuri dan Ibu Widarsih yang senantiasa memberikan dukungan berupa materil ataupun non materil berupa doa dan semangat. Kepada Kakak saya Widi Astuti NJ yang selalu memberikan dukungan dan semangat. Kepada seluruh dosen Jurusan Teknik Informatika UIN Maulana Malik Ibrahim Malang yang senantiasa mencurahkan ilmunya dan tidak lelah membimbing penulis sehingga mampu menuntaskan skripsi.

Kepada Dolpina dan sahabat Ucil saya dan orang-orang terdekat yang penulis sayangi baik terlibat secara langsung ataupun dalam bentuk dukungan untuk tidak hentinya berprogress sehingga mampu menyelesaikan skripsi ini. Terima kasih untuk semuanya.

KATA PENGANTAR

Puji Syukur kepada Allah SWT atas limpahan rahmat dan karunia serta hidayah-Nya, sehingga skripsi yang berjudul “Klasifikasi Kualitas Jeruk Lokal Menggunakan Metode *Neural Network* “ dapat terselesaikan dengan baik. Tujuan peneliti mengerjakan skripsi ini guna memenuhi syarat kelulusan dan mendapatkan gelar sarjana komputer (S.Kom) di program studi Teknik Informatika, Fakultas Sains dan Teknologi, Universitas Islam Maulana Malik Ibrahim Malang.

Selama penyusunan skripsi ini, penulis mendukung dukungan dari banyak pihak berupa dukungan, bantuan, bimbingan, serta arahan dari berbagai pihak. Oleh karena itu, dengan kerendahan hati penulis menyampaikan terima kasih kepada:

1. Prof. Dr. H. M. Zainuddin, MA selaku Rektor Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang.
2. Dr. Sri Harini, M.Si selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang.
3. Dr. Fachrul Kurniawan, M.MT selaku Ketua Prodi Teknik Informatika Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang
4. Dr. Cahyo Crysdiyan, M.Cs selaku Dosen Pembimbing I yang telah memberikan arahan, bimbingan, dan dorongan selama penulisan skripsi ini sehingga dapat menyelesaikan skripsi ini dengan tepat waktu.

5. Dr. Irwan Budi Santoso, M.Kom selaku Dosen Pembimbing II yang sudah membantu memberikan arahan, bimbingan, dan dorongan selama penulisan skripsi ini, sehingga dapat menyelesaikan skripsi ini dengan tepat waktu
6. Dr. Ir. Mokhamad Amin Hariyadi, M.T selaku Dosen Penguji 1 dan Ajib Hanani, M.T selaku Dosen Penguji II yang sudah memberikan saran terbaiknya dan sudah meluangkan waktunya untuk menguji penulis dalam skripsi ini.
7. Hani Nurhayati, M.T selaku dosen wali yang telah memberikan bimbingan dan motivasi selama penulis menjalani masa studi di Program Studi Teknik Informatika UIN Maulana Malik Ibrahim Malang.
8. Bapak, ibu, dan kakak penulis yang memberikan banyak dukungan, motivasi, doa, dan pengorbanan kepada penulis selama proses belajar hingga menyelesaikan program studi S1. Teman teman angkatan 2019 “ALIEN” Teknik Informatika yang memberikan dukung kepada penulis.
9. Bapak dan Ibu dosen dan staff Program Studi Teknik Informatika yang telah memberikan ilmu dan membantu proses perkuliahan hingga penyusunan skripsi.
10. Serta semua pihak yang tidak bisa penulis sebutkan yang telah berkontribusi dalam menyelesaikan skripsi ini sehingga dapat terselesaikan dengan baik.

Malang, 6 Juni 2024

Penulis

DAFTAR ISI

HALAMAN PENGAJUAN	ii
HALAMAN PERSETUJUAN	iii
HALAMAN PENGESAHAN	iv
PERNYATAAN KEASLIAN TULISAN	v
HALAMAN MOTTO	vi
HALAMAN PERSEMBAHAN	vii
KATA PENGANTAR	viii
DAFTAR ISI	x
DAFTAR GAMBAR	xii
DAFTAR TABEL	xiii
ABSTRAK	xiv
ABSTRACT	xv
مستخلص البحث	xvi
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Identifikasi Masalah	4
1.3 Tujuan Penelitian	5
1.4 Batasan Masalah	5
1.5 Manfaat Penelitian	6
BAB II STUDI PUSTAKA	7
2.1 Klasifikasi Kualitas Buah	7
2.2 Identifikasi Kualitas Jeruk	10
2.3 <i>Neural Network</i>	12
BAB III DESAIN DAN IMPLEMENTASI	14
3.1 Pengumpulan Data	14
3.2 Augmentasi Data	15
3.3 Perhitungan Nilai Diameter dan Tinggi dari Objek Citra	16
3.4 Desain Sistem	21
3.4.1 <i>Preprocessing</i> Data Gambar	21
3.4.2 <i>Preprocessing</i> Data Numerik	24
3.4.3 <i>Neural Network</i>	25
3.4.4 <i>Training Neural Network</i>	32
3.5 Desain Tampilan Aplikasi	34
3.6 Implementasi Sistem	35
BAB IV UJI COBA DAN PEMBAHASAN	39
4.1 Data Penelitian	39
4.2 Skenario Uji Coba	41
4.2.1 Perhitungan Model Sistem 1	42
4.2.2 Perhitungan Model Sistem 2	43
4.2.3 Perhitungan Model Sistem 3	44
4.3 Aplikasi Desktop dan Hardware	46
4.4 Menghitung Kinerja Sistem	48
4.5 Hasil Uji Coba	50

4.5.1 Sistem 1 (70:30).....	54
4.5.2 Sistem 1 (80:20).....	55
4.5.3 Sistem 2 (70:30).....	56
4.5.4 Sistem 2 (80:20).....	57
4.5.5 Sistem 3 (70:30).....	58
4.5.6 Sistem 3 (80:20).....	59
4.6 Pembahasan.....	60
BAB V KESIMPULAN DAN SARAN.....	66
5.1 Kesimpulan	66
5.2 Saran	67
DAFTAR PUSTAKA	

DAFTAR GAMBAR

Gambar 3. 1 Alur Perhitungan Nilai Diameter dan Tinggi.....	17
Gambar 3. 2 Desain Sistem.....	21
Gambar 3. 3 Alur <i>Preprocessing</i> Data Gambar.....	22
Gambar 3. 4 Contoh Hasil Join <i>Image</i> Sisi Buah Jeruk 1, 2, dan 3.	23
Gambar 3. 5 Gambar dengan Warna Dasar Merah (<i>Red</i>).....	24
Gambar 3. 6 Gambar dengan Warna Dasar Hijau (<i>Green</i>).....	24
Gambar 3. 7 Gambar dengan Warna Dasar Biru (<i>Blue</i>).....	24
Gambar 3. 8 Alur <i>Preprocessing</i> Data Numerik.....	24
Gambar 3. 9 Arsitektur NN pertama untuk klasifikasi kematangan buah jeruk ...	27
Gambar 3. 10 Flowchart Proses Training	33
Gambar 3. 11 Desain Tampilan Sistem.....	34
Gambar 3. 12 Implementasi Sistem	35
Gambar 3. 13 Hasil Output	38
Gambar 4. 1 Contoh data gambar asli jeruk kelas kematangan A	40
Gambar 4. 2 Contoh hasil augmentasi data dari kelas kematangan A.....	40
Gambar 4. 3 Contoh data yang tersimpan dalam format .csv	40
Gambar 4. 4 Data gambar yang digunakan untuk proses training	41
Gambar 4. 5 Interface utama program	47
Gambar 4. 6 Interface pengujian data yang tersimpan.....	47
Gambar 4. 7 Hardware tampak dalam dan komponen.....	48

DAFTAR TABEL

Tabel 2. 1 Kode Ukuran SNI 3165:2009	11
Tabel 2. 2 Kode Kelas Kematangan pada Penelitian	11
Tabel 2. 3 Kode Kelas Ukuran pada Penelitian	11
Tabel 3. 1 Kelas kematangan jeruk	14
Tabel 3. 2 Kelas Ukuran Jeruk	15
Tabel 3. 3 Detail Arsitektur <i>Hidden Layer</i> yang Digunakan	28
Tabel 4. 1 Model <i>Neural Network</i> Sistem 1	42
Tabel 4. 2 Model <i>Neural Network</i> Sistem 2	44
Tabel 4. 3 Model <i>Neural Network</i> Sistem 3	46
Tabel 4. 4 Hasil uji coba setiap sistem	51
Tabel 4. 5 Grafik hasil uji coba setiap sistem	52
Tabel 4. 6 Perhitungan <i>Confusion Matrix</i> Sistem 1 (70:30)	54
Tabel 4. 7 Perhitungan <i>Confusion Matrix</i> Sistem 1 (80:20)	55
Tabel 4. 8 Perhitungan <i>Confusion Matrix</i> Sistem 2 (70:30)	56
Tabel 4. 9 Perhitungan <i>Confusion Matrix</i> Sistem 2 (80:20)	57
Tabel 4. 10 Perhitungan <i>Confusion Matrix</i> Sistem 3 (70:30)	58
Tabel 4. 11 Perhitungan <i>Confusion Matrix</i> Sistem 2 (80:20)	59

ABSTRAK

Abdillah, Junaedi. 2024. **Klasifikasi Kualitas Jeruk Lokal Menggunakan Metode *Neural Network***. Skripsi. Program Studi Teknik Informatika. Fakultas Sains dan Teknologi. Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (1) Dr. Cahyo Crysdiyan, M.Cs (2) Dr. Irwan Budi Santoso, M.Kom.

Kata Kunci : *Neural Network, Klasifikasi, Kualitas, Jeruk Lokal*

Buah Jeruk merupakan buah yang memiliki nilai ekonomis tinggi dan dibudidayakan secara luas oleh petani sebagai sumber pendapatan utama. Namun budidaya jeruk di Indonesia masih lebih mengutamakan cara tradisional terutama pada tahap pasca panen. tingkat hasil dan mutu produk jeruk yang dijual masih rendah. Penelitian ini bertujuan membuat sistem yang mampu mengklasifikasikan kualitas jeruk berdasarkan tingkat kematangan dan keseragaman bentuk menurut standar SNI 3165:2009 menggunakan metode *Neural Network* (NN). *Input* proses menggunakan data citra berformat *RGB* pada 3 sisi permukaan jeruk yang berbeda 120 derajat melingkupi seluruh permukaan jeruk yang telah melewati *preprocessing* berupa *remove background*, *resize* (200x200), penggabungan 3 gambar (200x600), dan normalisasi gambar. *Input* proses juga menggunakan data numerik yang berisi data diameter, tinggi, dan berat jeruk yang telah dilakukan normalisasi. Kelas klasifikasi berjumlah 20 kelas yang berupa kombinasi dari kelas kematangan yang diwakili huruf A-E dan kelas ukuran yang diwakili angka 1-4. Uji coba sistem dibagi menjadi 3 yaitu sistem 1 – 3 dan masing masing sistem diuji dengan 2 kombinasi data training testing yang berbeda yaitu 70:30 dan 80:20. Setelah dilakukan pengujian pada sistem diperoleh performa terbaik pada sistem 3 (80:20) dengan nilai *accuracy* 77.5%, *precision* 80.8%, *recall* 80.8%, dan *f-measure* 80.8%. Performa yang cukup baik pada sistem 3 dipengaruhi oleh tingkat kompleksitas arsitektur yang cukup tinggi dengan total 17 *Hidden Layer* sehingga sistem mampu mengklasifikasikan kombinasi kelas kematangan dan kelas ukuran dengan baik.

ABSTRACT

Abdillah, Junaedi. 2023. **Classification of Local Orange Quality Using Neural Network Method**. Thesis. Department of Informatics Engineering, Faculty of Science and Technology, Maulana Malik Ibrahim Islamic State University Malang. Supervisor : (1) Dr. Cahyo Crysdian, M.Cs (2) Dr. Irwan Budi Santoso, M.Kom.

Orange fruit is a fruit with high economic value and is widely cultivated by farmers as a primary source of income. However, orange cultivation in Indonesia still primarily uses traditional methods, especially in the post-harvest stage. The yield and quality of oranges sold are still low. This study aims to create a system capable of classifying the quality of oranges based on ripeness level and shape uniformity according to the SNI 3165:2009 standard using the Neural Network (NN) method. The input process uses RGB image data from 3 different surfaces of the orange at 120-degree angles covering the entire surface of the orange, which has undergone preprocessing such as background removal, resizing (200x200), combining 3 images (200x600), and image normalization. The input process also uses numerical data containing the diameter, height, and weight of the orange, which has been normalized. The classification has 20 classes, which are combinations of ripeness classes represented by letters A-E and size classes represented by numbers 1-4. The system testing is divided into 3 systems (systems 1-3), and each system is tested with 2 different training and testing data combinations, namely 70:30 and 80:20. After testing the system, the best performance was obtained in system 3 (80:20) with an accuracy of 77.5%, precision of 80.8%, recall of 80.8%, and f-measure of 80.8%. The fairly good performance in system 3 is influenced by the high complexity of the architecture with a total of 17 hidden layers, allowing the system to classify the combination of ripeness and size classes well.

Keywords: *Neural Network, Classification, Quality, Local Orange*

مستخلص البحث

عبد الله، جنيدى. 2024. تصنيف جودة البرتقال المحلي باستخدام طريقة الشبكة العصبية. البحث الجامعي. قسم الهندسة المعلوماتية. كلية العلوم والتكنولوجيا بجامعة مولانا مالك إبراهيم الإسلامية الحكومية مالانج. المشرف الأول: د. جهيو كريسديان. المشرف الثاني: إيروان بودي سانتوسو، الماجستير.

الكلمات الرئيسية: شبكة عصبية، تصنيف، جودة، برتقال محلي.

فاكهة البرتقال هي فاكهة ذات قيمة اقتصادية عالية ويزرعها المزارعون على نطاق واسع كمصدر رئيسي للدخل. ومع ذلك، لا تزال زراعتها في إندونيسيا تعطي الأولوية للطرق التقليدية، خاصة في مرحلة ما بعد الحصاد. لا يزال مستوى العائد وجودة منتجات البرتقال المباع منخفضة. هدف هذا البحث إلى إنشاء نظام قادر على تصنيف جودة البرتقال بناء على مستوى النضج وتوحيد الشكل وفقا لمعيار SNI 3165: 2009 باستخدام طريقة الشبكة العصبية (NN). يستخدم إدخال العملية بيانات صورة RGB على 3 جوانب مختلفة من السطح البرتقالي 120 درجة تغطي السطح البرتقالي بالكامل الذي مر بالمعالجة المسبقة في شكل إزالة الخلفية، وتغيير الحجم (200 × 200)، ودمج 3 صور (600 × 200)، وتطبيع الصورة. استخدم إدخال العملية أيضا بيانات رقمية تحتوي على بيانات حول قطر وارتفاع ووزن البرتقال الذي تم تطبيعه. هناك 20 فئة تصنيف في شكل مجموعة من فئات النضج ممثلة بالحروف A-E وفئات الحجم ممثلة بالأرقام 1-4. تم تقسيم تجربة النظام إلى 3 أنظمة، وهي نظام 1 إلى 2 وتم اختبار كل نظام بثلاث مجموعات مختلفة من بيانات اختبار التدريب، وهي 70:30، 80:20. بعد الاختبار على النظام، تم الحصول على أفضل أداء في النظام 3 (80:20) بقيمة دقة 77.5%، ثبات 80.8%، استدعاء 80.8%، وقيمة ف 80.8%. يتأثر الأداء الجيد في نظام 3 بالمستوى العالي من التعقيد المعماري بإجمالي 17 طبقة مخفية بحيث يكون النظام قادرا على تصنيف مجموعة من فئات النضج وفئات الحجم بشكل جيد.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Buah Jeruk merupakan buah yang memiliki nilai ekonomis tinggi dan dibudidayakan secara luas oleh petani sebagai sumber pendapatan (Hanif, 2020). Besarnya minat pasar terhadap produk jeruk sejalan dengan luasnya area budidaya jeruk di Indonesia, namun budidaya jeruk di Indonesia masih lebih mengutamakan cara tradisional terutama pada tahap pasca panen. tingkat hasil dan mutu produk jeruk yang dijual masih rendah (Abbas & Suhaeti, 2016).

Penggunaan teknologi pasca panen, khususnya tahap pemilahan di Indonesia, menghadirkan sejumlah kendala dan tantangan karena alasan berikut (1) alasan pribadi seperti ekonomi dan sosial (2) alasan terkait organisasi takut menyebabkan hal yang buruk atau tidak memuaskan, (3) alasan ketidaktahuan terkait inovasi dan pengenalan teknologi (4) alasan menjaga etika terhadap budaya yang sudah lama ada (Abbas & Suhaeti, 2016).

Hambatan terkait hal tersebut di atas juga didukung oleh penelitian yang dilakukan di Kabupaten Karo Sumatera Utara, grading manual produksi jeruk di tingkat petani, dan proses nonklasifikasi berdasarkan kualitas warna dan rasa, sehingga keseragaman bentuk buah dan kualitasnya tidak terjamin, sehingga pemasok ke pasar seperti supermarket yang ingin mempertahankan kualitasnya harus melakukan klasifikasi ulang (Purba & Purwoko, 2019). Hal ini disebabkan berbagai keterbatasan yang terkait dengan perkembangan teknologi dan

penggunaan alat yang masih tradisional dan sederhana serta rendahnya penguasaan teknologi baru di bidang ini (Dalimunthe, 2019).

Permasalahan klasifikasi jeruk telah dibahas pada beberapa jurnal terkait, seperti penelitian (Syaputra, 2022) yang mengurutkan kualitas berdasarkan diameter, berat dan ketebalan kulit menggunakan *Fuzzy K-Nearest Neighbor* atau penelitian (Yanto et al., 2021) penilaian kematangan jeruk manis berbasis citra dengan perhitungan parameter kecerahan warna menggunakan metode *Neural Network*. Dan penelitian (Gunawan, 2013) mengklasifikasikan jeruk Kintamani berdasarkan karakteristik warna dan ukuran dengan menggunakan metode jarak *Euclidean*. Namun, studi yang relevan memeringkat kualitas buah jeruk lokal berdasarkan pengolahan citra untuk mendeteksi kematangan dan keseragaman ukuran sesuai dengan SNI (Standar Nasional Indonesia) 3165:2009 belum ada.

Pemilahan hasil panen merupakan proses yang cukup penting untuk meningkatkan mutu dan nilai jual hasil pertanian, karena jeruk yang telah matang, bentuk yang sempurna dan ukuran yang seragam harganya akan lebih mahal daripada jeruk yang ada dipasaran yang memiliki cacat fisik atau tidak matang sempurna. Selain mempengaruhi harga jual, pemilahan juga dapat mencegah penyebaran pembusukan atau penularan ke produk lain selama penyimpanan atau pengangkutan untuk diperdagangkan (Utama, 2015).

Klasifikasi adalah cara pengelompokan objek ke dalam kelompok tertentu berdasarkan karakteristik yang dimiliki oleh masing-masing pengklasifikasi, proses ini dapat dilakukan secara manual atau menggunakan teknologi seperti kecerdasan buatan (Wibawa et al., 2018). Salah satu metode klasifikasi

kecerdasan buatan yang digunakan adalah *Neural Network* yang diimplementasikan dalam penelitian (Purnama, 2020) untuk menentukan tingkat kematangan alpukat dengan akurasi 94,7% dibandingkan dengan *SVM* yang hanya mencapai 94,4%. Jaringan saraf tiruan memiliki keunggulan karena pembelajaran terjadi secara berulang melalui node yang saling berhubungan untuk menciptakan sistem yang dapat mengklasifikasikan berbagai jenis objek secara konsisten dan akurat (Wibawa et al., 2018).

Penelitian yang dilakukan memiliki tujuan dapat mengklasifikasikan jeruk berdasarkan ciri kualitas yang dimiliki, sehingga dapat membantu masyarakat terutama para petani dalam tahap pasca panen. Pada penelitian ini setiap jeruk memiliki ciri khusus yang dapat diidentifikasi ke dalam suatu kelas tertentu untuk dapat diprediksi kategori kelas kualitasnya, sejalan pada surah Muhammad ayat 30 sebagai berikut:

وَأَوْ نَشَاءُ لَأَرَيْنَكُمُهُمْ فَلَعَرَفْتَهُمْ بِسِيمَاهُمْ يَوْمَ نَعْرِفُهُمْ فِي لَحْنِ الْقَوْلِ وَاللَّهُ يَعْلَمُ أَعْمَالَكُمْ

“Seandainya Kami berkehendak, niscaya Kami menunjukkan mereka kepadamu (Nabi Muhammad) sehingga engkau benar-benar dapat mengenali mereka melalui tanda-tandanya. Engkau pun benar-benar akan mengenali mereka melalui nada bicarannya. Allah mengetahui segala amal perbuatanmu” (Q.S Muhammad/47:30).

Menurut buku yang disusun oleh Ishaq, (2013) terjemahan Tafsir oleh Ibnu Katsir, ayat tersebut menjelaskan bahwa setiap orang memiliki tanda atau ciri untuk dapat dikenali dalam ayat tersebut dicontohkan berupa ciri melalui nada bicarannya yang mana Allah mengetahui segala amal perbuatan makhluknya. Hal ini saling terkait dengan penelitian yang dilakukan dimana ciri suatu jeruk dapat

diklasifikasikan juga dengan mengenali warna kulit yang menunjukkan tingkat kematangan serta mengenali kualitas bentuk dari ukuran yang dimiliki.

Dalam penelitian ini, metode jaringan saraf digunakan untuk mengklasifikasikan kualitas jeruk. Klasifikasi mutu yang dimaksud adalah untuk menentukan kematangan dan keseragaman bentuk menurut standar SNI 3165:2009. Klasifikasi berdasarkan pengolahan citra pada 3 permukaan jeruk yang berbeda 120 derajat sekaligus melingkupi seluruh permukaan jeruk, dan pengolahan data numerik yang berisi data diameter, tinggi dan berat jeruk. Hasil tahap klasifikasi berupa kombinasi dari kode kelas dari kelas kematangan yang diwakili huruf A-E dan kelas ukuran yang diwakili angka 1-4 yang merepresentasikan keseragaman ukuran buah jeruk. Pemilihan metode *Neural Network* dikarenakan kemampuannya dalam mempelajari objek sebagai *input* jaringan (Adhi et al., 2016) sehingga cocok untuk mengolah data multivariasi berbagai tipe data seperti yang digunakan dalam penelitian ini.

1.2 Identifikasi Masalah

1. Seberapa baik tingkat akurasi, presisi, *f-measure*, dan *recall* dari penerapan metode *Neural Network* untuk permasalahan klasifikasi kualitas jeruk berdasarkan citra digital, berat, diameter, dan tinggi?
2. Apa saja faktor yang dapat berpengaruh pada performa algoritma *Neural Network* dalam pengklasifikasian kualitas jeruk berdasarkan citra dan bentuk?

1.3 Tujuan Penelitian

1. Menghitung tingkat akurasi, presisi, *f-measure*, dan *recall* dari penerapan metode *Neural Network* untuk permasalahan klasifikasi kualitas jeruk berdasarkan citra digital, berat, diameter, dan tinggi.
2. Menganalisis faktor-faktor yang mempengaruhi kinerja algoritma *Neural Network* pada klasifikasi kualitas buah jeruk berdasarkan citra digital, berat, diameter dan tinggi dengan membuat beberapa model arsitektur yang berbeda, dan setiap arsitektur dianalisis dengan *confusion matrix* untuk mencari faktor yang mempengaruhinya.

1.4 Batasan Masalah

1. Jenis buah jeruk yang diteliti tergolong dalam jenis jeruk keprok (*Citrus Reticulata*).
2. Data pada tiap buah jeruk terdiri dari gambar 3 sisi buah jeruk yang berbeda tiap 120 derajat serta bentuk fisik jeruk (diameter, tinggi, dan berat) dengan catatan data yang digunakan telah melalui tahap pencucian.
3. Data primer yang digunakan dibuat secara mandiri melalui pengambilan data jeruk dari pasar ataupun petani jeruk secara langsung. Dan terdapat proses augmentasi data untuk memperkaya jumlah data dengan tujuan agar sistem mampu mengenali objek lebih baik dengan jumlah dataset yang sangat terbatas karena keterbatasan waktu dan sumber daya yang dimiliki. Total jumlah data asli yang diperoleh adalah 130 buah jeruk dan dilakukan proses augmentasi pada data training sebanyak 10x. Untuk percobaan digunakan dua kombinasi data *training* dan data *testing* yaitu 70:30 dan 80:20.

4. Proses pengkelasan dataset didasarkan pada SNI (Standar Nasional Indonesia) 3165:2009 untuk klasifikasi kelas A-C dengan tambahan klasifikasi kelas D untuk jeruk yang setengah matang dan E untuk jeruk yang busuk, mentah atau tidak layak konsumsi serta kelas ukuran 1-4 yang ditambahkan parameter tinggi agar mampu mengenali keseragaman bentuk lebih baik.

1.5 Manfaat Penelitian

Manfaat penelitian ditujukan untuk pengembang mesin pertanian pasca panen pada proses pemilahan kualitas buah yaitu:

1. Membantu tahap pasca panen bagi para petani dalam pemilahan kualitas jeruk berdasarkan standar ekspor untuk meningkatkan mutu dan nilai jual produk.
2. Membantu penelitian dalam bidang riset pengembangan mesin pemilah kualitas jeruk berdasarkan citra dan keseragaman bentuk untuk proses klasifikasi.
3. Sebagai referensi penelitian selanjutnya untuk pengembangan alat pertanian yang memanfaatkan sistem kecerdasan buatan.

BAB II

STUDI PUSTAKA

2.1 Klasifikasi Kualitas Buah

Klasifikasi kualitas buah adalah proses mengelompokkan buah berdasarkan ciri fisik yang terlihat. Ciri fisik tersebut dapat berupa warna kulit buah yang mempresentasikan tingkat kematangan ataupun kerusakan. Implementasi *deep learning* untuk mengklasifikasikan kualitas buah memiliki tantangan tersendiri, karena terdapat banyak parameter yang mempengaruhi hasil kinerjanya (Denata et al., 2021). Umumnya klasifikasi kualitas buah dapat dilakukan menggunakan *artificial intelligence* yang berbasis *machine learning* seperti *KNN* atau *deep learning* seperti *Neural Network* dengan mempertimbangkan parameter berupa tekstur kulit dan warna kulitnya dengan buah matang memiliki ciri kulit halus, tipis, berkilau, dan warna cenderung terang (Syaputra, 2022). Selain berdasarkan penilaian citra, pengklasifikasian dapat dilakukan dengan mengkombinasikan bentuk, warna, dan ukuran untuk menghasilkan keluaran klasifikasi yang lebih akurat dan memiliki keseragaman yang baik (Hartanto, 2017).

Klasifikasi kualitas buah dengan pendekatan *deep learning* pernah dilakukan oleh (Dwi & Wicaksono, 2022), penelitian tersebut bertujuan untuk mengklasifikasikan kematangan, mutu dan varietas pisang menggunakan jaringan saraf tiruan berdasarkan karakteristik warna dan bentuk. Data yang digunakan berasal dari Perkebunan Pisang Cavendish di Kabupaten Banyumas, Jawa Tengah. Terdapat 9 jenis pisang yang digunakan untuk klasifikasi dengan parameter kematangan, kematangan dan pisang mentah. Langkah perhitungan

klasifikasi adalah menerapkan filter untuk mendapatkan ekstraksi ciri dari objek citra untuk mendapatkan tepi horizontal dan vertikal. Hasil perhitungan akan dilanjutkan pada lapisan tersembunyi untuk mendapatkan hasil klasifikasi. Penelitian ini berhasil membuat sistem klasifikasi jenis dan tingkat kematangan dengan tingkat akurasi yang cukup baik yaitu 98,7%.

Kemudian Fharadila et al., (2019) melakukan penelitian untuk mengklasifikasikan kematangan nanas menggunakan jaringan syaraf tiruan. Penelitian ini menggunakan 100 citra data nanas madu rimbo panjang dengan 70 citra untuk pelatihan dan 30 citra untuk pengujian. Klasifikasi didasarkan pada dua kelas, matur dan mentah dengan parameter *input* ke jaringan saraf berupa nilai *RGB* (merah, hijau, biru) dari objek gambar. Perhitungan terbaik pada penelitian ini menggunakan parameter *epoch* 1000, *learning rate* 0.1, target 0.001 dan *Hidden Layer* 5 dengan akurasi tertinggi mencapai 100%.

Pada penelitian Mazen & Nashat, (2019) mengklasifikasikan pisang menjadi 4 kelas dengan metode jaringan syaraf tiruan dengan menerapkan ekstraksi ciri berupa kekasaran, kontras, arah, rentang nilai kroma dan RF. Model yang digunakan menggunakan 10 lapisan tersembunyi dengan fungsi aktivasi sigmoid karena sederhananya fungsi turunan dan transformasi yang baik. Lapisan keluaran adalah 4 *neuron* tergantung pada jumlah 4 lapisan matang yang dipelajari. Dataset yang digunakan sebanyak 300 citra dengan dataset *training* sebesar 70%. Penelitian ini juga membandingkan *Neural Network* dengan algoritma klasifikasi lainnya seperti *SVM*, *Naïve Bayes*, *KNN*, pohon keputusan dan analisis

diskriminan, dengan hasil kinerja optimal yang dicapai oleh JST dengan akurasi 97,75%.

Terdapat juga penelitian yang menerapkan integrasi *artificial intelligence* dengan mesin penyortir berdasarkan *image processing* yang dilakukan oleh (Siskandar et al., 2020), penelitian tersebut merancang alat sortir kematangan jeruk dan tomat berdasarkan *image processing*. Deteksi fitur pemrosesan citra untuk menilai kematangan berdasarkan nilai Merah/Hijau yang diperoleh dari citra kamera dengan menerapkan logika fuzzy pada pengambilan keputusan. Prinsip pengoperasian mesin akan melakukan proses penginderaan yang dilakukan oleh *Raspberry Pi* melalui screenshot, hasil proses *Raspberry Pi* ditransmisikan ke Arduino Mega untuk mengeksekusi perintah kontrol ke servo dan langkah untuk analisis memisahkan objek yang terdeteksi berdasarkan usia. Hasil akhir alat dapat memberikan akurasi 100% berdasarkan percobaan pada 10 buah jeruk dan 10 buah tomat.

Penelitian lain yang mengintegrasikan antara *artificial intelligence* dan mesin penyortir juga dilakukan oleh (Abdul Aziz et al., 2021), penelitian tersebut melakukan penelitian terkait integrasi penyortir cabai berbasis warna dengan kecerdasan buatan menggunakan metode *ANN (Artificial Neural Network)*. Data yang digunakan hingga 20 frame dan 40 frame, masing-masing frame menghasilkan akurasi berkisar antara 80-85%, dimana data citra diperoleh melalui CCD kamera. *Preprocessing* dilakukan dengan melakukan *cropping* citra untuk menghilangkan *background* menggunakan ROI (*region of interest*), dilanjutkan dengan *resize* citra, konversi nilai *RGB* ke CIE, ketajaman citra dan ekstraksi

warna. Pada bagian proses arsitektur menggunakan 10 *Hidden Layer* dengan 2 *Layer output* sebagai klasifikasi merah atau hijau dengan 20 *epoch*. Pengoperasian mesin membaca data melalui CCD kamera dan mengirimkannya ke komputer melalui Arduino untuk proses klasifikasi dengan metode ANN, hasilnya dikirim kembali ke Arduino untuk menggerakkan dan mengontrol motor stepper untuk memindahkan cabai sesuai kategorinya.

2.2 Identifikasi Kualitas Jeruk

Berdasarkan ketentuan SNI (Standar Nasional Indonesia) 3165:2009 terkait kualitas jeruk berjenis keprok (*Citrus Reticulata*) dapat dinilai berdasarkan kondisi visual seperti warna kulit yang mengindikasikan kematangan dan tingkat kerusakan pada buah, serta penggunaan parameter ukuran dari jeruk seperti diameter dan berat. Pada standar SNI tersebut jeruk digolongkan menjadi 3 kelas yang terdiri dari kelas super, A, dan B.

Kelas super merupakan golongan kelas tertinggi dengan kondisi fisik hanya sedikit memiliki kerusakan pada permukaan kulit ataupun bentuk.

Kelas A digolongkan jika memiliki sedikit penyimpangan permukaan kulit ataupun bentuk yang diakibatkan proses ketika pembentukan buah atau bekas karena mekanis Ketika panen atau proses pasca panen. Toleransi kerusakan yang diperbolehkan maksimal 10% dari luas total keseluruhan dan tidak mempengaruhi kualitas daging buah.

Kelas B memiliki ciri yang hampir mirip dengan kelas A dengan batas toleransi kecacatan dan penyimpangan bentuk maksimal 15% dari total luas permukaan dan tidak mempengaruhi kualitas daging buah.

Kemudian ukuran buah jeruk dibagi menjadi 4 golongan berdasarkan diameter. Seperti yang tercantum dalam tabel 2.1.

Tabel 2. 1 Kode Ukuran SNI 3165:2009

Kode Ukuran	Diameter (milimeter)
1	> 70
2	61 - 70
3	51 - 60
4	40 - 50

Sedangkan identifikasi yang diangkat pada penelitian ini merupakan penambahan dari standar SNI 3165:2009 yang sudah ada yang bertujuan agar semua kelas dapat teridentifikasi dengan baik sesuai tujuan penelitian yang diangkat yaitu mengklasifikasikan kualitas jeruk pasca panen. Penambahan kelas tersebut adalah kelas D yang merupakan jeruk setengah matang dan E yang merupakan jeruk masih mentah atau busuk, serta mengubah penamaan kelas kematangan dari standar SNI yaitu Super, A, B menjadi kelas A, B, C. Serta penambahan satu parameter pada penilaian kelas ukuran yaitu nilai tinggi dimana perbedaan diameter dan tinggi umumnya berada pada kisaran 5 – 15 mm.

Tabel 2. 2 Kode Kelas Kematangan pada Penelitian

Kelas (Kematangan)	Tingkat Kematangan
A	Matang
B	Matang
C	Matang
D	Setengah Matang
E	Busuk, Mentah

Tabel 2. 3 Kode Kelas Ukuran pada Penelitian

Kode Ukuran	Diameter (milimeter)	Tinggi (milimeter)	Berat (gram)
1	> 70	> 65	> 150
2	61 - 70	55 – 65	101 – 150
3	51 - 60	45 – 55	51 – 100
4	40 - 50	35 – 45	< 50

2.3 Neural Network

Neural Network merupakan salah satu metode *artificial intelligence* yang bekerja dengan menirukan kerja otak manusia dari beberapa sudut 1) Pengetahuan pada *Network* merupakan hasil pembelajaran dari lingkungan, 2) Kekuatan koneksi antar unit disebut *synaptic weight*, difungsikan menyimpan pengetahuan yang telah dipelajari jaringan tersebut, 3) Toleransi kesalahan yang dimiliki metode ini mirip dengan sistem biologis, ketika nilai kesalahan cukup kecil maka proses penyesuaian parameter berbobot untuk memperoleh *output* bergantung pada harga pembobotan interkoneksi sel yang selanjutnya akan dihentikan (Amrizal & Aini, 2013). Secara umum metode ini mengikuti konsep *Neural* sistem pada organisme yang menjadi dasar kerja otak dengan tujuan untuk “mempelajari” suatu tugas dengan memperhitungkan model yang sebagian besar tanpa perlu memodifikasi *guide-line* pada tugas secara eksplisit. Pengaplikasian *Neural Network* sangat luas terutama untuk menyelesaikan tugas sulit yang membutuhkan banyak parameter atau kelas termasuk pada Prediksi dan klasifikasi (Yousuf et al., 2021)

Arsitektur *Neural Network* terdiri kedalam beberapa lapisan yang memiliki proses masing masing. Lapisan pertama adalah *input Layer* yang menerima masukan data dari sistem bisa berupa data numerik, citra, ataupun teks. Lapisan kedua merupakan *Hidden Layer* suatu lapisan yang setiap unitnya tersembunyi dan hasil prosesnya tidak bisa langsung diamati atau dibaca. Dan terakhir adalah *output Layer* lapisan yang berisi hasil proses yang dipecahkan oleh *Neural Network* (Amrizal & Aini, 2013).

Penerapan *Neural Network* untuk klasifikasi jeruk dilakukan oleh Setiawan Wibisono et al., (2019) melakukan penelitian klasifikasi kualitas buah jeruk berdasarkan warna serta ukurannya menggunakan metode Fuzzy C-Means dan *Neural Network* untuk klasifikasi kualitas. Data penelitian diperoleh dari pedagang jeruk di wilayah Kabupaten Semarang dengan jumlah 50 foto citra jeruk yang dibagi kedalam 25 jeruk mentah dan 25 jeruk matang berjenis jeruk siam jawa. Perhitungan dilakukan dengan proses *resize* yang dilanjutkan mencari nilai *RGB* dan mencari nilai centroid untuk proses klasifikasi, dengan hasil akhir metode *Neural Network* mampu memperoleh akurasi sebesar 83%.

Kemudian Penelitian Yanto et al., (2023) melakukan proses klasifikasi jeruk manis untuk deteksi jeruk tersebut bagus atau busuk menggunakan algoritma NN (*Neural Network*) yang didasarkan pada citra gambar *RGB* (*Red, Green, Blue*). Dataset citra jeruk diambil secara langsung menggunakan kamera Cannon EOS 1200D dengan resolusi 18 Mp berjumlah 100 data. Tahap *preprocessing* dilakukan dengan mengubah ukuran citra menjadi 64x64 piksel dan kernel menjadi 3. Pada tahap proses menggunakan 8 filter yang terdiri dari *Image Input Layer, Convolutional Layer, Batch Normalization Layer, ReLu Layer, Max Pooling Layer, Fully Connected Layer, Softmax Layer, Classification Layer* dengan kernel sebesar 3x3 menggunakan *epoch* berjumlah 50. Akurasi yang diperoleh baik dengan tingkat akurasi 96% *training* dan 92% *testing*.

BAB III

DESAIN DAN IMPLEMENTASI

3.1 Pengumpulan Data

Penelitian ini membahas tentang klasifikasi kualitas jeruk berdasarkan ciri fisik berupa warna permukaan kulit, tinggi, diameter, dan berat untuk menghasilkan klasifikasi dalam mengidentifikasi tingkat kematangan dan keseragaman bentuk yang membutuhkan data berupa citra dan numerik. Data yang dibutuhkan diperoleh dari pengamatan langsung dan pengambilan citra buah jeruk di pasar tradisional serta hasil panen petani jeruk di Kabupaten Magelang, Jawa Tengah. Klasifikasi kualitas jeruk pada penelitian ini dibagi menjadi 5 kelas kematangan dan 4 kelas kode ukuran yang dijabarkan pada table 3.1 dan 3.2. Data yang digunakan berupa gambar buah jeruk dengan *background* yang telah dihilangkan dengan objek berjumlah 3 gambar yang diambil dari setiap sisi yang berbeda tiap 120 derajat. Kemudian data numerik yang digunakan berupa ukuran diameter, tinggi, dan berat dari buah jeruk yang diperoleh dari pengukuran objek secara langsung.

Tabel 3. 1 Kelas kematangan jeruk

Kelas	Tingkat Kematangan	Keterangan
A	Matang	Kondisi fisik hanya sedikit memiliki kerusakan pada permukaan kulit ataupun bentuk
B	Matang	Memiliki sedikit penyimpangan permukaan kulit ataupun bentuk yang diakibatkan proses ketika pembentukan buah atau bekas karena mekanis. Toleransi kerusakan yang diperbolehkan maksimal 10%.
C	Matang	Batas toleransi kecacatan dan penyimpangan bentuk maksimal 15%.
D	Setengah Matang	Buah jeruk yang belum matang sempurna yang ditandai dengan jumlah total permukaan kulit yang berwarna orange dan hijau hampir sama.
E	Busuk, Mentah	Buah jeruk mentah ditandai dengan permukaan kulit yang keseluruhan hampir berwarna hijau gelap. Dan buah jeruk busuk ditandai terdapat bagian berwarna kecoklatan atau kehitaman yang

		menandakan adanya pembusukan atau tanda lain berwarna putih yang berarti memiliki penyakit atau jamur.
--	--	--

Tabel 3. 2 Kelas Ukuran Jeruk

Kode Ukuran	Diameter (milimeter)	Tinggi (milimeter)	Berat (gram)
1	> 70	> 65	> 150
2	61 - 70	55 – 65	101 – 150
3	51 - 60	45 – 55	51 – 100
4	40 - 50	35 – 45	< 50

3.2 Augmentasi Data

Augmentasi data merupakan suatu teknik untuk memperbanyak jumlah data dengan melakukan beberapa perubahan kecil. Ketika data yang digunakan terbatas namun bentuk model yang dibangun kompleks dan membutuhkan banyak data untuk mencapai tingkat akurasi yang baik.

Augmentasi data dilakukan pada data training, dimana sebelum proses augmentasi data dilakukan terdapat proses *split* data training dan testing terlebih dahulu dengan perbandingan 70:30, dan 80:20. Teknik augmentasi data yang digunakan pada penelitian ini ada dua yaitu augmentasi data untuk gambar dan data numerik. Untuk proses augmentasi data gambar dilakukan dengan membuat kombinasi ulang gambar pada 3 sisi buah pada kelas yang sama serta memperbanyak gambar buah jeruk dengan merubah sedikit bentuk data asli dengan cara merotate dalam rentang -90 dan 90 derajat serta mem-flip gambar. Sedangkan proses augmentasi data numerik dilakukan dengan melakukan penilaian ulang data diameter, tinggi, berat, dan klasifikasi.

Teknik augmentasi data numerik tersebut memiliki beberapa aturan agar data yang dihasilkan relevan berdasarkan data yang sudah ada dan standar ukuran

sesuai SNI (Standar Nasional Indonesia) 3165:2009, berikut ini ketentuan prosesnya,

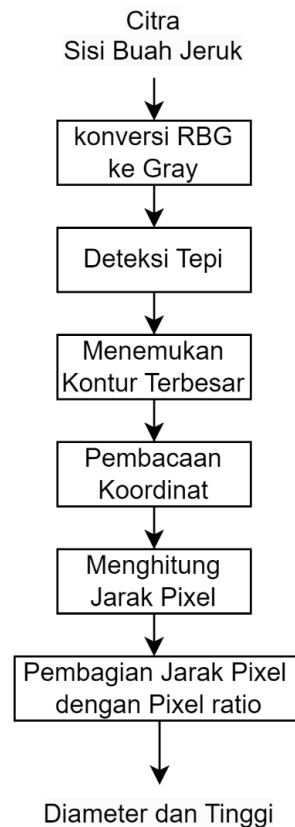
- a. Data diameter dibuat dalam rentang 40 – 79 milimeter (Batas minimal bawah kode ukuran 4 dan batas atas kode ukuran 1)
- b. Data tinggi diperoleh dari pengurangan data berat dengan nilai rentang 5 – 15 milimeter (Margin perbedaan tinggi dan diameter jeruk Keprok)
- c. Data berat dibuat berdasarkan 4 kondisi,
 1. Kondisi 1, Ketika diameter berada pada rentang 71 – 79, maka berat dibuat pada rentang 151 – 200 gram
 2. Kondisi 2, Ketika diameter berada pada rentang 61 – 70, maka berat dibuat pada rentang 101 – 150 gram
 3. Kondisi 3, Ketika diameter berada pada rentang 51 – 60, maka berat dibuat pada rentang 51 – 100 gram
 4. Kondisi 4, Ketika diameter berada pada rentang 40 – 50, maka berat dibuat pada rentang 30 – 50 gram

Penilaian ulang klasifikasi didasarkan pada diameter yang dimiliki dan kualitas kematangan buah jeruk.

3.3 Perhitungan Nilai Diameter dan Tinggi dari Objek Citra

Perhitungan *pixel ratio* digunakan untuk memperoleh nilai diameter dan tinggi objek nyata berdasarkan gambar yang ditangkap kamera. Proses menghitung nilai diameter dan tinggi pada objek gambar dilakukan dengan menemukan nilai a dan b yang merupakan nilai koordinat sisi terluar objek untuk memperoleh nilai jarak *pixel* diameter dan tinggi serta nilai Tengah pada objek.

Nilai jarak tersebut dibagi dengan nilai *pixel ratio* untuk memperoleh ukuran nyata. Proses lengkap dapat dilihat pada diagram 3.1 berikut,



Gambar 3. 1 Alur Perhitungan Nilai Diameter dan Tinggi

3.3.1 Konversi *RGB* ke *Gray*

Proses ini dilakukan untuk memudahkan dalam deteksi tepi gambar. Algoritma yang digunakan adalah *Luminosity* yaitu mengalikan nilai *RGB* dengan faktor skalar (0.299, 0.587, 0.114) suatu nilai yang memberikan intensitas luminansi piksel kedalam citra *grayscale*, faktor skalar dirancang sesuai dengan sensitivitas mata manusia terhadap komponen warna. Nilai faktor skalar yang umum digunakan untuk perkalian tersebut adalah 0.299 untuk *Red*, 0.587 untuk

Green, dan 0.114 untuk *blue* yang diperoleh dari buku yang ditulis Gonzalez & Woods (2018).

$$Gray \begin{pmatrix} x \\ y \end{pmatrix} = 0.299 R(x, y) + 0.587 G(x, y) + 0.114 B(x, y) \quad (3.1)$$

Keterangan :

- x = koordinat *pixel* pada posisi horizontal
- y = koordinat *pixel* pada posisi vertikal
- R = nilai *pixel* merah pada gambar
- G = nilai *pixel* hijau pada gambar
- B = nilai *pixel* biru pada gambar

3.3.2 Deteksi Tepi

Proses deteksi tepi digunakan untuk menentukan kontur atau sisi terluar objek. Proses ini dilakukan dengan algoritma *Gaussian Blur* terlebih dahulu untuk menghaluskan gambar sehingga memudahkan menemukan kontur objek yang sebenarnya dan *Canny Edge Detection* untuk menemukan sisi terluar objek. Persamaan yang digunakan untuk filter *Gaussian Blur* pada gambar mengikuti penelitian dari Mohapatra (2019) sebagai berikut,

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (3.2)$$

Keterangan :

- $G(x, y)$ = Nilai piksel hasil perhitungan filter Gaussian
- σ = Parameter untuk mengontrol standar deviasi distribusi Gaussian
- e = Konstanta Euler

Deteksi tepi dengan *canny* mengimplementasikan teknik dalam menemukan intensitas gradien dalam gambar. Ketika terdapat perbedaan nilai antar *pixel* yang besar maka posisi tersebut akan diberikan nilai jalur tepi dengan *hysteresis*. Berdasarkan penelitian Mohapatra (2019) perhitungan deteksi tepi dengan *Canny*

terdapat beberapa langkah, pertama menghitung gradien citra dan magnitudo gradien, persamaan tersebut sebagai berikut,

$$G_x^{\square} = \frac{\partial I}{\partial x}, G_y^{\square} = \frac{\partial I}{\partial y} \quad (3.3)$$

$$E(x, y) = \sqrt{(G_x^{\square}(x, y))^2 + (G_y^{\square}(x, y))^2} \quad (3.4)$$

Keterangan :

$E(x, y)$ = Intensitas tepi koordinat (x,y)
 G_x^{\square} = Turunan parsial citra terhadap x
 G_y^{\square} = Turunan parsial citra terhadap y
 I = Fungsi yang diambil turunannya

Langkah selanjutnya yaitu menghitung arah gradien yang kemudian digunakan untuk menentukan orientasi tepi pada tiap titik gambar. Persamaan tersebut adalah,

$$\theta(x, y) = \arctan\left(\frac{G_y^{\square}(x, y)}{G_x^{\square}(x, y)}\right) \quad (3.5)$$

Keterangan :

$\theta(x, y)$ = Arah sudut gradien (x,y)

3.3.3 Menemukan Kontur Terbesar dan Pembacaan Koordinat Sisi

Ketika garis tepi pada objek sudah ditemukan dengan perhitungan tepi *Canny* maka dilakukan proses perulangan untuk memperoleh kontur terluar dari objek. Pembacaan sisi terluar dilakukan dengan *library cv2.minAreaRect* untuk menentukan kotak terkecil yang melingkupi kontur yang telah ditemukan dan *library cv2.BoxPoints* untuk menemukan koordinat keempat sudut dari kotak.

3.3.4 Menemukan Jarak *Pixel*

Proses selanjutnya menghitung jarak *pixel* diameter dan tinggi objek yang terdapat beberapa langkah. Proses paling awal adalah menentukan koordinat titik tengah dari keempat sisi dengan titik koordinat tiap sudut yang telah diketahui dengan mengikuti perhitungan geometri sederhana pada buku yang ditulis Jacobs (1987) dengan persamaan,

$$(midX(x_a, y_a), midY(x_b, y_b)) = \left(\frac{x_a + x_b}{2}, \frac{y_a + y_b}{2} \right) \quad (3.6)$$

Nilai titik tengah yang telah diketahui kemudian digunakan untuk menghitung jumlah *pixel* diameter dan tinggi objek dengan menggunakan persamaan *euclidean* yaitu,

$$PixelDiameter = \sqrt{(midXL(x_a, y_a) - midXR(x_b, y_b))^2 + (midYL(x_a, y_a) - midYR(x_b, y_b))^2} \quad (3.7)$$

$$PixelTinggi = \sqrt{(midXB(x_a, y_a) - midXT(x_b, y_b))^2 + (midYB(x_a, y_a) - midYT(x_b, y_b))^2} \quad (3.8)$$

Keterangan :

$midX, midY$	= Nilai tengah dari dua koordinat (x, y)
$midXL, midYL$	= Nilai tengah sisi kiri
$midXR, midYR$	= Nilai tengah sisi kanan
$midXB, midYB$	= Nilai tengah sisi bawah
$midXT, midYT$	= Nilai tengah sisi atas

3.3.5 Menghitung ukuran diameter dan tinggi objek

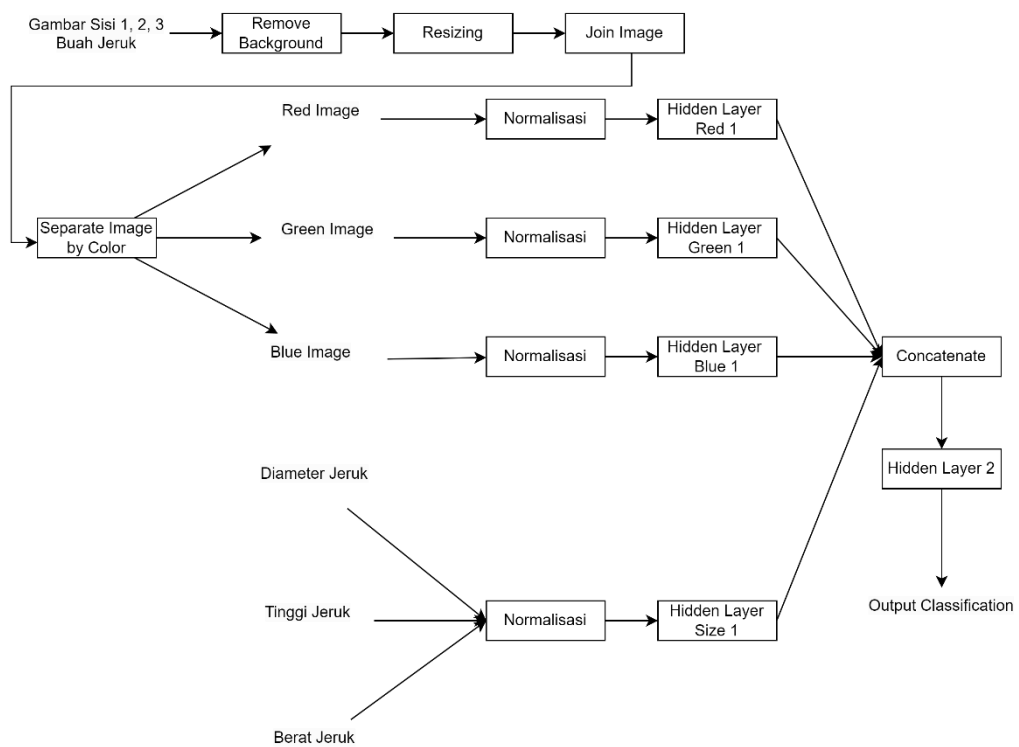
Hasil perhitungan jumlah *pixelDiameter* dan *pixelTinggi* dikonversi kedalam jarak sebenarnya dengan membaginya dengan *pixel_ratio*. Persamaannya sebagai berikut,

$$Diameter = pixel_diameter / pixel_ratio \quad (3.9)$$

$$Tinggi = pixel_tinggi / pixel_ratio \quad (3.10)$$

3.4 Desain Sistem

Gambar 3.2 merupakan desain sistem kecerdasan buatan yang digunakan pada penelitian ini.



Gambar 3. 2 Desain Sistem

3.4.1 Preprocessing Data Gambar

Terdapat beberapa proses tahapan yang diterapkan dalam *preprocessing* data gambar sisi buah jeruk yang dijabarkan pada gambar 3.3.



Gambar 3. 3 Alur *Preprocessing* Data Gambar

A. *Remove Background*

Gambar buah jeruk yang digunakan memiliki warna latar yang terang cenderung putih namun tidak sepenuhnya putih. Sehingga perlu dilakukan proses menghilangkan warna latar agar tidak mempengaruhi hasil. Proses *remove background* dilakukan dengan *library* python yaitu *rembg*.

B. *Resize*

Setelah warna latar belakang diperoleh proses selanjutnya melakukan *resize* dengan ukuran 200×200 *pixel* yang merupakan *pixel* terkecil data yang ditemukan untuk menjaga kualitas data agar tidak kehilangan detail informasi setelah proses *resize*. Proses ini diperlukan agar gambar mampu diolah oleh sistem. Metode yang digunakan adalah interpolasi *linear* yang didasarkan pada penelitian S. Parsania & V.Virparia (2015) dalam proses perhitungannya. Interpolasi *linear* digunakan untuk *resize* gambar dengan memperkirakan nilai piksel baru dari garis lurus antar dua nilai piksel terdekat. Perhitungan yang dilakukan dengan meletakkan *pixel* yang akan dihitung dalam koordinat (x,y) dimana *pixel* tersebut berada diantara empat *pixel* tetangga dengan koordinat (x_1,y_1) , (x_2,y_1) , (x_1,y_2) , (x_2,y_2) dengan memenuhi $x_1 \leq x \leq x_2$ dan $y_1 \leq y \leq y_2$ yang kemudian dihitung dalam persamaan, $f(x_1,y_1)$, $f(x_2,y_1)$, $f(x_1,y_2)$, dan $f(x_2,y_2)$ merupakan nilai piksel dari empat *pixel* tetangga pada koordinat yang dihitung dan x_1 , y_1 , x_2 , dan y_2 merupakan koordinat *pixel* tetangga.

$$f(x, y) = f(x_1, y_1) * (x_2 - x) * (y_2 - y) + f(x_2, y_1) * (x - x_1) * (y_2 - y) + f(x_1, y_2) * (x_2 - x) * (y - y_1) + f(x_2, y_2) * (x - x_1) * (y - y_1) / ((x_2 - x_1) * (y_2 - y_1)) \quad (3.11)$$

C. Join Image

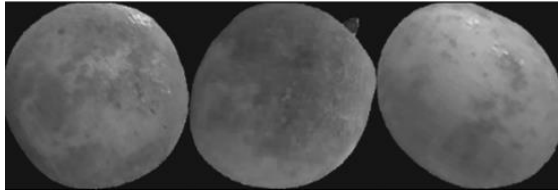
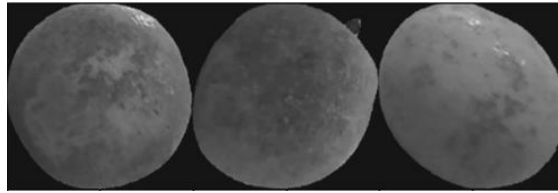
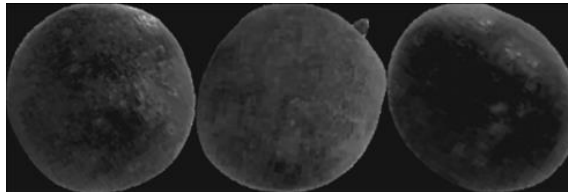
Setelah dilakukan proses *resize* pada tiap data gambar asli kedalam ukuran 200x200 *pixel*, kemudian masing masing data gambar sisi 1, 2, dan 3 dipasangkan kembali berdasarkan kombinasi ketika proses augmentasi. Hasil pengabungan dari 3 sisi tersebut menghasilkan ukuran gambar dengan ukuran *pixel* 200x600. Pengabungan gambar bertujuan agar data 3 sisi gambar diproses menjadi satu kesatuan entitas buah jeruk ketika diolah dalam *Neural Network*.



Gambar 3. 4 Contoh Hasil Join *Image* Sisi Buah Jeruk 1, 2, dan 3.

D. Memisah Nilai *Red, Green, Blue*

Proses memisah citra nilai berdasarkan jenis warna dasar sebelum di proses kedalam *Neural Network* bertujuan agar setiap warna dapat diproses pada *Layer input* yang berbeda. Hal demikian dilakukan agar setiap fitur warna dasar dapat dikenali lebih baik dan lebih akurat ketika diproses yang memungkinkan untuk menghasilkan model yang mampu mengklasifikasikan setiap fitur unik gambar kedalam kelas tertentu.

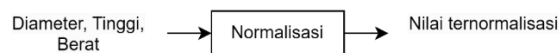
Gambar 3. 5 Gambar dengan Warna Dasar Merah (*Red*)Gambar 3. 6 Gambar dengan Warna Dasar Hijau (*Green*)Gambar 3. 7 Gambar dengan Warna Dasar Biru (*Blue*)

E. Normalisasi Gambar

Normalisasi merupakan proses untuk mengolah data dengan tujuan mengubah skala nilai menjadi nilai yang relatif sama berada pada rentang 0-1. Tahap normalisasi diterapkan pada nilai setiap array *pixel* dimana nilai *pixel* dibagi dengan 255 (nilai terbesar piksel gambar).

$$f(x) = \frac{x}{255} \quad (3.12)$$

3.4.2 Preprocessing Data Numerik

Gambar 3. 8 Alur *Preprocessing* Data Numerik

Normalisasi merupakan proses untuk mengolah data dengan tujuan mengubah skala nilai menjadi nilai yang relatif sama berada pada rentang 0-1.

Tahap normalisasi diterapkan pada data numerik yang berisi diameter, tinggi, dan berat dari buah jeruk. Metode yang diterapkan merupakan min-max normalization, metode min-max normalization akan merubah nilai terkecil pada fitur menjadi 0, nilai terbesar pada fitur menjadi 1, dan nilai lain akan diubah pada rentang 0 hingga 1. Berikut persamaan dari metode min-max *normalization* yang didasarkan pada penelitian Patro & sahu (2015),

$$f(x) = \frac{x - \min(x)}{\max(x) - \min(x)} \quad (3.13)$$

3.4.3 *Neural Network*

A. *Arsitektur Neural Network*

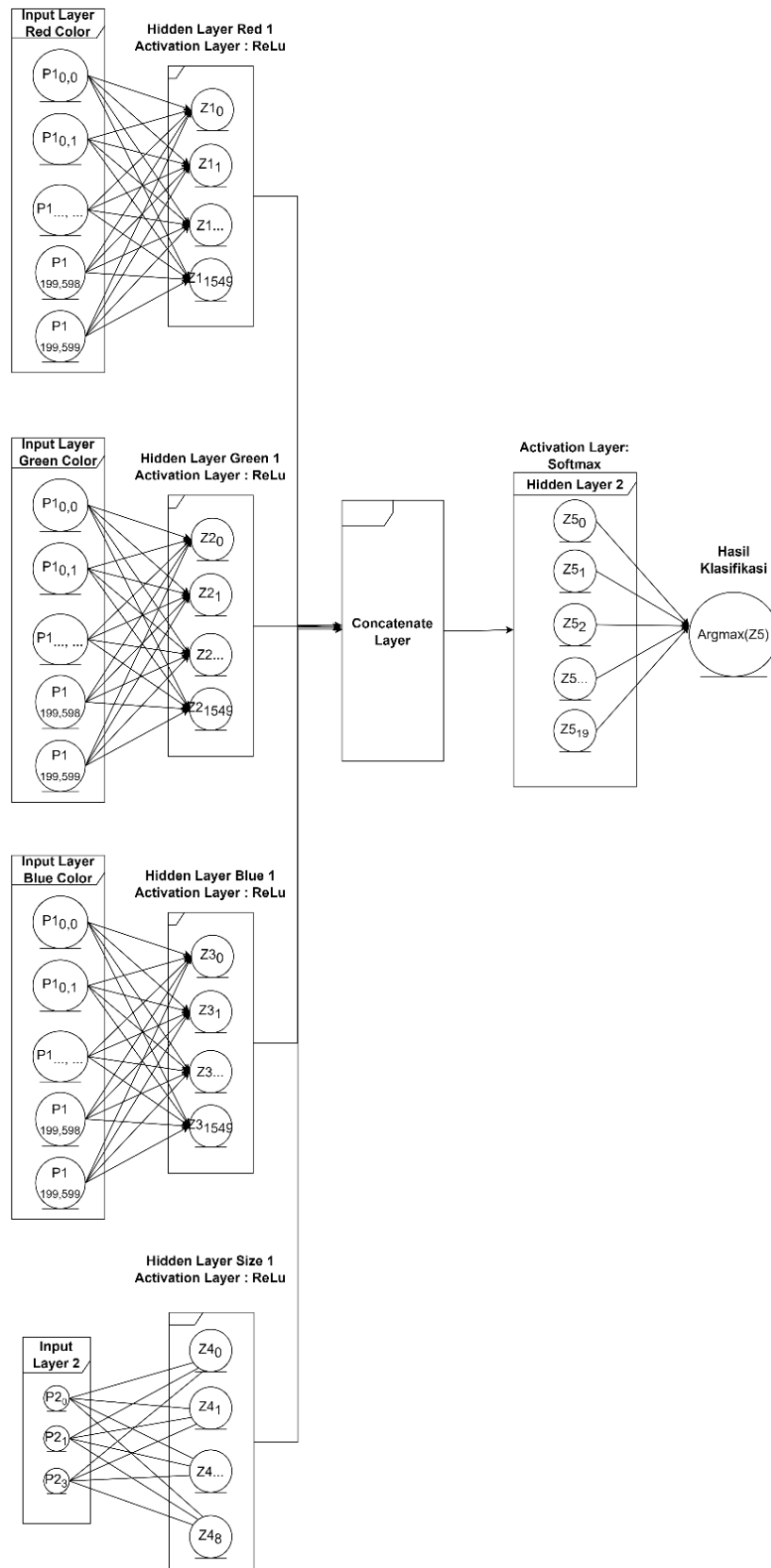
Arsitektur *Neural Network* terdiri dari beberapa bagian dasar

1. *Input*, masukan yang diperlukan untuk pembelajaran atau pengenalan objek,
2. Satu set *synapses* atau jaringan yang saling terhubung, setiap *synapses* memiliki berat atau kekuatan yang terdiri dari rentang nilai positif hingga negatif.
3. *Adder*, menjumlahkan masukan untuk setiap sinyal masukan yang dibaca pada tiap *neuron*.
4. Fungsi aktivasi, pembatasan amplitudo keluaran dari *neuron*. Rentang yang dimiliki biasanya dinormalisasi dalam interval tertutup [0,1] atau [-1,1].
5. *Output*, hasil keluaran dari pengenalan objek berdasarkan tahap tahapan sebelumnya.

Singkatnya proses *Neural Network* diawali dengan satu set *synapses input* membawa aktivasi dari *neuron* lain dilanjutkan pembacaan oleh unit pengolah

dengan menerapkan fungsi aktivasi non-*linear* dan diakhiri *line output* mengirim keluaran ke *neuron* lain (Santoso, 2021).

Arsitektur dasar *Neural Network* yang digunakan ditunjukkan pada Gambar 3.5, terdapat 4 *input Layer* yang diterapkan pada arsitektur NN. *Input Layer* 1-3 merupakan data citra dari buah jeruk yang telah dipisah menjadi nilai *Red*, *Green*, dan *Blue* agar model mampu mempelajari fitur warna lebih baik dan *input Layer* 4 merupakan data numerik yang berisi diameter, tinggi, dan berat. *Input Layer* 1-4 diproses pada *Hidden Layer* masing masing yang kemudian *Hidden Layer* tersebut digabung dalam *concentrate layer*. *Concentrate Layer* dihubungkan kedalam *output Layer* yang terdiri dari 20 node sesuai jumlah klasifikasi yang digunakan. Hasil proses tersebut adalah klasifikasi kelas buah jeruk yang dibagi menjadi 20 kelas yang terdiri dari kombinasi kelas kematangan A-E dan keseragaman bentuk 1-4.



Gambar 3. 9 Arsitektur NN pertama untuk klasifikasi kematangan buah jeruk

B. Input Layer

Input Layer 1-3 berisi nilai piksel dari 3 gambar yang telah dilakukan proses *preprocessing*, digabung, dan dipisah nilai RGB-nya dengan resolusi 200 x 600 *pixel*. Kemudian *input Layer* ke-empat berisi 3 data numerik yang mewakili nilai dari diameter (mm), tinggi (mm), dan berat (gram) buah jeruk.

C. Hidden Layer

Hidden Layer yang digunakan merupakan *fully connected Layer* dimana setiap *neuron* terhubung dengan *neuron* sebelumnya. *Hidden Layer* yang digunakan berjumlah 5 dengan detail yang dapat dilihat pada tabel 3.3. Terlihat pada tabel 3.3 bahwa *Hidden Layer - 1* memiliki 20 *neuron* yang merupakan kombinasi dari kelas kematangan (5 kelas) dan kelas ukuran (4 kelas) jadi *output* yang dihasilkan berupa kombinasi kelas dengan contoh jika jeruk dengan kelas kematangan A dengan ukuran kelas 2 akan menghasilkan nilai A2.

Tabel 3. 3 Detail Arsitektur *Hidden Layer* yang Digunakan

<i>Hidden Layer</i>	Jumlah <i>Neuron</i>	Fungsi Aktivasi
<i>Hidden Layer Red - 1</i>	1549	ReLU
<i>Hidden Layer Green - 1</i>	1549	ReLU
<i>Hidden Layer Blue - 1</i>	1549	ReLU
<i>Hidden Layer Size - 1</i>	8	ReLU
<i>Hidden Layer - 1</i>	20	Softmax

Untuk perhitungan setiap *neuron* pada tiap *Hidden Layer* tertera pada persamaan dibawah:

$$z_s^1 = f \left(\sum_{i=0}^{l-1} w_{i,s}^1 p_{i,s}^1 + b_s^1 \right) \quad (3.14)$$

$$z_s^2 = f \left(\sum_{i=0}^{l-1} w_{i,s}^2 p_{i,s}^2 + b_s^2 \right) \quad (3.15)$$

$$z_s^3 = f \left(\sum_{i=0}^{l-1} w_{i,s}^3 p_{i,s}^3 + b_s^3 \right) \quad (3.16)$$

$$z_s^4 = f \left(\sum_{i=0}^{l-1} w_{i,s}^4 p_{i,s}^4 + b_s^4 \right) \quad (3.17)$$

$$z_s^5 = g \left(\sum_{i=0}^{l-1} w_{i,s}^5 c_{i,s}^1 + b_s^5 \right) \quad (3.18)$$

Keterangan:

- s = indeks untuk jumlah *neuron* pada *layer*
- i = indeks untuk jumlah *neuron* pada *Layer* sebelumnya
- I = total jumlah *neuron* pada *Layer* sebelumnya
- $p_{i,s}^1$ = nilai *input Layer* yang menuju *neuron* $z_{s,1}^x$
- $c_{i,s}^1$ = nilai dari concatenate *Layer* yang menuju *neuron* $z_{s,1}^x$
- b_s^x = bias ke - s yang berada pada p_s^x dan $z_{s,1}^x$ diinisialisasi nilai bilangan pecahan kecil.
- $w_{i,s}^x$ = bobot p_i^x pada *neuron* $z_{s,1}^x$ diinisialisasi nilai bilangan pecahan kecil
- z_s^x = *Hidden Layer* ke-x dengan jumlah *neurons*

Huruf f pada persamaan 3.14 hingga 3.17 mewakili dari fungsi aktivasi

ReLU. Berikut ini persamaan dari fungsi aktivasi ReLU yang digunakan

$$f_{i,j}(z_{i,j}) = \begin{cases} 0 & \text{for } x \leq 0 \\ x & \text{for } x > 0 \end{cases} \quad (3.19)$$

Sedangkan huruf g pada persamaan 3.18 merupakan fungsi aktivasi softmax dengan persamaan,

$$g_{i,j}(z_{i,j}) = \frac{e^{z_{i,j}}}{\sum_{j=0}^{l-1} e^{z_{i,j}}} \quad (3.20)$$

Keterangan :

- $f_{i,j}$ = fungsi aktivasi ReLU
- $g_{i,j}$ = fungsi aktivasi Softmax
- z = nilai perhitungan dari *Hidden layer*

D. Output Layer

Output diperoleh dari hasil perhitungan softmax yang berisi kategori dengan probabilitas tertinggi dari semua kategori, Ketika semua probabilitas dijumlahkan maka bernilai 1, persamaan pencarian nilai terbesar dengan persamaan:

$$\tau(x) = \max (z_0^6, \dots , z_{19}^6) \quad (3.21)$$

Keterangan:

$\tau(x)$ = Kelas dengan probabilitas tertinggi

$z_{i,j}^6$ = Hasil probabilitas tiap kategori dari *Hidden Layer* terakhir

E. Loss Function

Nilai *Loss* merupakan suatu nilai metrik yang mengukur kemampuan model dalam mengkategorikan atau memprediksi nilai yang sebenarnya. Perhitungan *Loss* berfungsi untuk mengoptimalkan model sehingga memiliki nilai *loss* yang kecil. Fungsi *loss* yang digunakan pada model ini adalah *Categorical Crossentropy Loss* dengan tujuan klasifikasi multikelas dengan tiap sampel termasuk ke dalam satu dari beberapa kelas yang ada.

$$L_i = - \sum_j y_{i,j} \log (\hat{y}_{i,j}) \quad (3.22)$$

Keterangan :

L_i = Nilai *Loss*

$y_{i,j}$ = Nilai target

$\hat{y}_{i,j}$ = Nilai hasil Prediksi

F. Adam Optimizer

Adaptif Momentum atau yang lebih dikenal Adam Optimizer merupakan optimizer yang umum digunakan dengan menggabungkan optimizer RMSProp dan perhitungan momentum dari SGD. Hasil pembaharuan parameter adaptif

menggunakan Adam memungkinkan model dapat beradaptasi lebih cepat pada parameter yang signifikan dan melambat pada parameter yang kurang penting.

$$m_i = \beta_1 \cdot m_{i-1} + (1 - \beta_1) \cdot g_i \quad (3.23)$$

$$v_i = \beta_2 \cdot v_{i-1} + (1 - \beta_2) \cdot g_i^2 \quad (3.24)$$

$$\hat{m}_i = \frac{m_i}{1 - \beta_1^i} \quad (3.25)$$

$$\hat{v}_i = \frac{v_i}{1 - \beta_2^i} \quad (3.26)$$

$$\theta_{i+1} = \theta_i - \frac{\alpha \cdot m_i}{\sqrt{\hat{v}_i} + \epsilon} \quad (3.27)$$

Keterangan :

θ_i	= Parameter model
g_i	= Gradien Parameter terhadap <i>loss function</i>
α	= <i>learning rate</i>
β_1 dan β_2	= <i>Smoothing factors</i> untuk gradien dan gradien kuadrat
m_i dan v_i	= moment estimation untuk gradien dan gradien kuadrat
\hat{m}_i dan \hat{v}_i	= moment estimation yang disesuaikan dengan bias
ϵ	= nilai kecil untuk mencegah pembagian dengan nol

G. Batch Normalization

Batch Normalization adalah suatu teknik normalisasi yang terjadi diantara *Layer* dari *Neural Network* sepanjang *mini-batch* yang bertujuan meningkatkan kecepatan pelatihan dan stabilitas dari suatu model. Cara kerja dari metode ini adalah menghitung rata rata dan standar deviasi dari tiap fitur pada *mini batch*, dimana tiap elemen *input* *mini-batch* dinormalisasi dari hasil perhitungan tersebut. Setelah diperoleh hasil normalisasi maka diubah menjadi skala dan dipindahkan dalam parameter *scaling* dan *shifting* untuk memberikan model fleksibilitas dalam mempelajari transformasi yang optimal dari *input* yang telah dinormalisasi.

$$\hat{x}_{ij} = \frac{x_{ij} - \mu}{\sqrt{\sigma^2 - \epsilon}} \quad (3.28)$$

$$y = \gamma \hat{x}_{ij} - \beta_{ij} \quad (3.29)$$

Keterangan :

\hat{x} = Standardization (hasil normalisasi)

x = *Input* dari *batch*

μ = Rata rata *batch* dari x

σ = standar deviasi dari x

ϵ = nilai kecil untuk mencegah pembagian dengan nol

y = hasil *scaling* dan *shifting*

γ = parameter skala

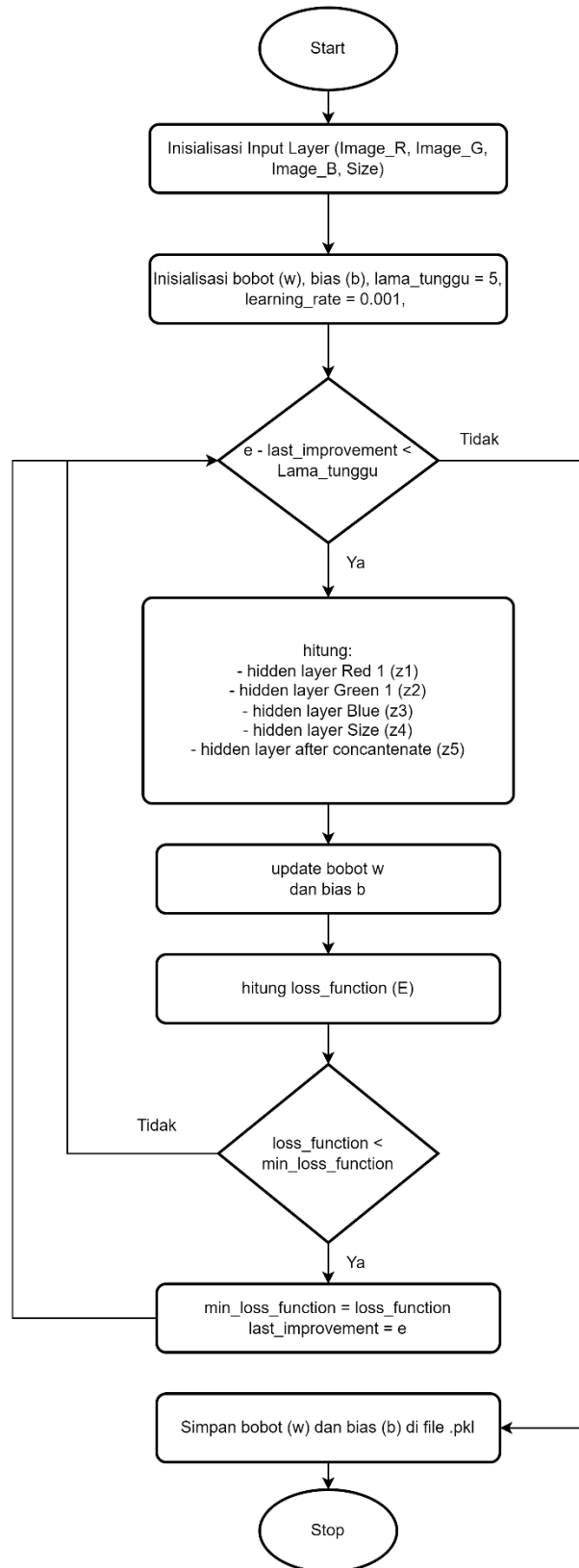
β = parameter pergeseran

3.4.4 *Training Neural Network*

Training merupakan cara sistem belajar berdasarkan data *training*, sehingga mampu mengenali bentuk dan perbedaan nilai antar kelas untuk melakukan proses klasifikasi. Proses ini untuk mengoptimalkan bobot. Diagram alur proses *training* dapat dilihat pada gambar 3.10.

Data proses yang digunakan pada proses *training* adalah gambar citra jeruk yang terdiri dari 3 gambar dari tiap sisi dengan *input* berupa nilai *RGB* serta nilai data numerik berupa diameter, tinggi, dan berat. Target atau *ground truth* berupa kelas kualitas yang terdiri dari kelas A hingga kelas E dan kelas ukuran yang terdiri dari kelas 1 – 4 yang sudah dilakukan penilaian dimana hasil perhitungan merupakan kombinasi kelas dimana kelas A1 bernilai 19 hingga E4 bernilai 0.

Tahap selanjutnya melakukan inisialisasi variabel yang diperlukan yaitu *learning rate*, *epoch*, dan nilai toleransi. Setelah itu terdapat inisialisasi bobot dan bias awal.



Gambar 3. 10 Flowchart Proses Training

3.5 Desain Tampilan Aplikasi

Desain tampilan sistem terdiri dari 6 form *input* yang terdiri dari 3 form *input* file gambar untuk tiap sisi jeruk dan 3 *input* berupa data angka untuk berat, diameter, dan tinggi buah jeruk. Tampilan tersebut memiliki satu tombol “proses” untuk mengolah semua data dan hasilnya akan ditampilkan pada form *output* berupa hasil klasifikasi dan deskripsi terkait klasifikasi tersebut. Desain secara umum seperti gambar 3.11.

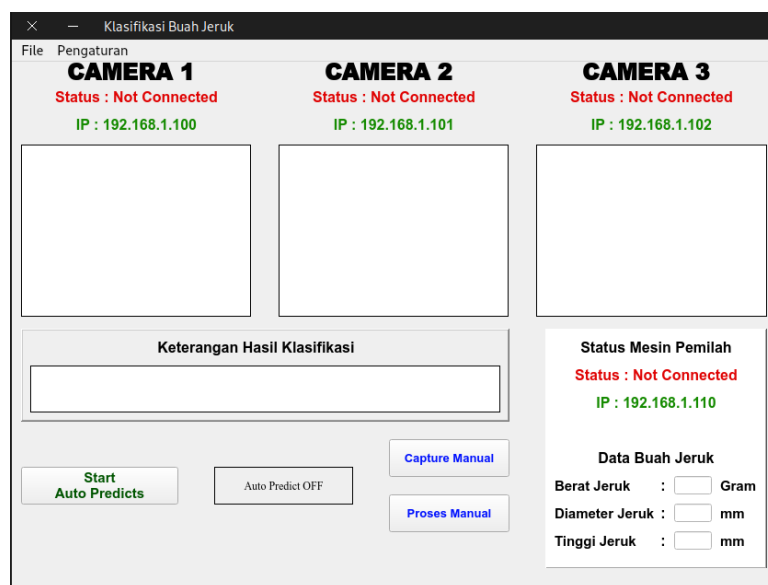
Tampilan GUI Data Monitor

<p>Gambar Sisi Jeruk 1</p> <div style="border: 1px solid gray; border-radius: 15px; width: 100%; height: 100%; display: flex; align-items: center; justify-content: center; margin: 5px;"> <p>Image 1</p> </div> <div style="border: 1px solid gray; border-radius: 10px; width: 100%; padding: 5px; margin-top: 5px;"> <p>Berat Jeruk (Saat Pengukuran)</p> </div>	<p>Gambar Sisi Jeruk 2</p> <div style="border: 1px solid gray; border-radius: 15px; width: 100%; height: 100%; display: flex; align-items: center; justify-content: center; margin: 5px;"> <p>Image 2</p> </div> <div style="border: 1px solid gray; border-radius: 10px; width: 100%; padding: 5px; margin-top: 5px;"> <p>Diameter Jeruk (Saat Pengukuran)</p> </div>	<p>Gambar Sisi Jeruk 3</p> <div style="border: 1px solid gray; border-radius: 15px; width: 100%; height: 100%; display: flex; align-items: center; justify-content: center; margin: 5px;"> <p>Image 3</p> </div> <div style="border: 1px solid gray; border-radius: 10px; width: 100%; padding: 5px; margin-top: 5px;"> <p>Tinggi Jeruk (Saat Pengukuran)</p> </div>
<div style="border: 1px solid gray; border-radius: 10px; width: 100%; padding: 5px; margin: 5px auto;"> <p>Proses</p> </div>		
<div style="border: 1px solid gray; border-radius: 10px; width: 100%; padding: 5px; margin: 5px auto;"> <p>Hasil Klasifikasi</p> <p>Deskripsi Klasifikasi</p> </div>		
<div style="display: flex; align-items: center; margin-bottom: 5px;"> <div style="font-size: 20px; margin-right: 5px;">📅</div> <p>Senin, 10 April 2023</p> </div> <div style="display: flex; align-items: center;"> <div style="font-size: 20px; margin-right: 5px;">🕒</div> <p>20:18:00</p> </div>		

Gambar 3. 11 Desain Tampilan Sistem

3.6 Implementasi Sistem

Implementasi sistem klasifikasi kualitas buah jeruk ditampilkan pada gambar 3.12.



Gambar 3. 12 Implementasi Sistem

Gambar 3.12 adalah *interface* utama pada tampilan sistem yang berfungsi membaca data masukan baru atau menampilkan data yang telah tersimpan sebelumnya. Pada *interface* utama terdapat proses membaca masukan baru, proses, dan *output*. Sistem klasifikasi ini akan memproses data masukan utama berupa satu buah jeruk dengan beberapa jenis *input* berupa 3 gambar setiap sisi dan data ukuran (diameter, berat, dan tinggi).

Sistem sudah terintegrasi dengan model *Neural Network* namun sebelum masukan utama diprediksi jenis kategorinya harus dilakukan tahap *preprocessing* terlebih dahulu yang telah dijelaskan pada sub-bab 3.3 konversi *pixel per matrix* untuk mengetahui nilai diameter tinggi dan sub-bab 3.4 *preprocessing* data gambar dan numerik.

```

import random
import traceback

import cv2
import numpy as np
import pandas as pd
from src.config import config_file, proses_file, proses_ukuran
from src.config.proses_ukuran import pixel_ratio
import src.config.lib.ANN as nn
import pickle
from sklearn.model_selection import train_test_split

read_csv_all = pd.read_csv(config_file.file_csv_training)
read_csv_not, read_csv = train_test_split(pd.read_csv(config_file.file_csv_training), test_size=0.2, random_state=0)
try:
    if dari_file:
        frame_nobg_hasil_1 = gambar[0]
        frame_nobg_hasil_2 = gambar[1]
        frame_nobg_hasil_3 = gambar[2]
        hasil_berat = data[0]
        hasil_tinggi = data[1]
        hasil_diameter = data[2]
    else:
        frame_nobg_hasil_1 = proses_file.proses_gambar(gambar[0])
        frame_nobg_hasil_2 = proses_file.proses_gambar(gambar[1])
        frame_nobg_hasil_3 = proses_file.proses_gambar(gambar[2])

        diameter1, tinggi1 = proses_ukuran.size_object(image=frame_nobg_hasil_1, pixelsPerMetric=pixel_ratio(data[2]))
        diameter2, tinggi2 = proses_ukuran.size_object(image=frame_nobg_hasil_2, pixelsPerMetric=pixel_ratio(data[1]))
        diameter3, tinggi3 = proses_ukuran.size_object(image=frame_nobg_hasil_3, pixelsPerMetric=pixel_ratio(data[3]))

        hasil_tinggi = np.array([tinggi1, tinggi2, tinggi3]).astype(int)
        hasil_diameter = np.array([diameter1, diameter2, diameter3]).astype(int)

        hasil_tinggi = np.max(hasil_tinggi)
        hasil_diameter = np.max(hasil_diameter)

        hasil_berat = proses_ukuran.hasil_berat(int(data[0]))

    berat_all = read_csv_all['Berat (gr)'].values
    berat_all = np.array(berat_all).astype(int)
    berat = (int(hasil_berat) - np.min(berat_all)) / (np.max(berat_all) - np.min(berat_all))
    diameter_all = read_csv_all['Diameter (mm)'].values
    diameter_all = np.array(diameter_all).astype(int)
    diameter = (int(hasil_diameter) - np.min(diameter_all)) / (np.max(diameter_all) - np.min(diameter_all))
    tinggi_all = read_csv_all['Tinggi (mm)'].values
    tinggi_all = np.array(tinggi_all).astype(int)
    tinggi = (int(hasil_tinggi) - np.min(tinggi_all)) / (np.max(tinggi_all) - np.min(tinggi_all))

    data_predict_ukuran = np.column_stack((berat, diameter, tinggi))
    data_predict_ukuran = np.array(data_predict_ukuran).reshape(1, 3)

    data_image_all = np.zeros((200, 600, 3), dtype='float16')
    dimension_x = 200
    dimension_y = 200
    data_image_all[0:200, 0:200] = cv2.resize(
        proses_file.crop_image(np.array(frame_nobg_hasil_1)),
        (dimension_x, dimension_y), interpolation=cv2.INTER_LINEAR)
    data_image_all[0:200, 200:400] = cv2.resize(
        proses_file.crop_image(np.array(frame_nobg_hasil_2)),
        (dimension_x, dimension_y), interpolation=cv2.INTER_LINEAR)
    data_image_all[0:200, 400:600] = cv2.resize(
        proses_file.crop_image(np.array(frame_nobg_hasil_3)),
        (dimension_x, dimension_y), interpolation=cv2.INTER_LINEAR)

    data_image_all = np.array(data_image_all).astype('float16') / 255
    R_data, G_data, B_data = cv2.split(data_image_all)
    # data_image_all = data_image_all.reshape(1, 200, 600, 3)

    berat_test = read_csv['Berat (gr)'].values
    diameter_test = read_csv['Diameter (mm)'].values
    tinggi_test = read_csv['Tinggi (mm)'].values

    berat_test = (berat_test - np.min(berat_test)) / (np.max(berat_test) - np.min(berat_test))
    diameter_test = (diameter_test - np.min(diameter_test)) / (np.max(diameter_test) - np.min(diameter_test))
    tinggi_test = (tinggi_test - np.min(tinggi_test)) / (np.max(tinggi_test) - np.min(tinggi_test))
    data_fisik = np.column_stack((berat_test, diameter_test, tinggi_test))
    data_fisik = np.array(data_fisik)

```

Ketika data sudah dimasukkan kedalam sistem, tahap pertama yang harus dilakukan adalah melakukan normalisasi menggunakan data yang ada. Tahap ini sangat penting untuk menyamakan skala bentuk data yang digunakan dalam proses *training*, agar masukan dapat diproses dan dikategorikan dengan baik.

```

for epoch in range(1):
    R1_dense.set_parameters(weights=loaded_model["R1_dense"][0], biases=loaded_model["R1_dense"][1])
    R1_dense.forward(R_test)
    R1_activation.forward(R1_dense.output)
    R2_dense.set_parameters(weights=loaded_model["R2_dense"][0], biases=loaded_model["R2_dense"][1])
    R2_dense.forward(R1_batch.forward(R1_activation.output))
    R2_activation.forward(R2_dense.output)
    R3_dense.set_parameters(weights=loaded_model["R3_dense"][0], biases=loaded_model["R3_dense"][1])
    R3_dense.forward(R2_batch.forward(R2_activation.output))
    R3_activation.forward(R3_dense.output)
    R4_dense.set_parameters(weights=loaded_model["R4_dense"][0], biases=loaded_model["R4_dense"][1])
    R4_dense.forward(R3_batch.forward(R3_activation.output))
    R4_activation.forward(R4_dense.output)

    G1_dense.set_parameters(weights=loaded_model["G1_dense"][0], biases=loaded_model["G1_dense"][1])
    G1_dense.forward(G_test)
    G1_activation.forward(G1_dense.output)
    G2_dense.set_parameters(weights=loaded_model["G2_dense"][0], biases=loaded_model["G2_dense"][1])
    G2_dense.forward(G1_batch.forward(G1_activation.output))
    G2_activation.forward(G2_dense.output)
    G3_dense.set_parameters(weights=loaded_model["G3_dense"][0], biases=loaded_model["G3_dense"][1])
    G3_dense.forward(G2_batch.forward(G2_activation.output))
    G3_activation.forward(G3_dense.output)
    G4_dense.set_parameters(weights=loaded_model["G4_dense"][0], biases=loaded_model["G4_dense"][1])
    G4_dense.forward(G3_batch.forward(G3_activation.output))
    G4_activation.forward(G4_dense.output)

    B1_dense.set_parameters(weights=loaded_model["B1_dense"][0], biases=loaded_model["B1_dense"][1])
    B1_dense.forward(B_test)
    B1_activation.forward(B1_dense.output)
    B2_dense.set_parameters(weights=loaded_model["B2_dense"][0], biases=loaded_model["B2_dense"][1])
    B2_dense.forward(B1_batch.forward(B1_activation.output))
    B2_activation.forward(B2_dense.output)
    B3_dense.set_parameters(weights=loaded_model["B3_dense"][0], biases=loaded_model["B3_dense"][1])
    B3_dense.forward(B2_batch.forward(B2_activation.output))
    B3_activation.forward(B3_dense.output)
    B4_dense.set_parameters(weights=loaded_model["B4_dense"][0], biases=loaded_model["B4_dense"][1])
    B4_dense.forward(B3_batch.forward(B3_activation.output))
    B4_activation.forward(B4_dense.output)

    data_fisik_dense_1.set_parameters(weights=loaded_model["data_fisik_dense_1"][0], biases=loaded_model["data_fisik_dense_1"][1])
    data_fisik_dense_1.forward(fisik_test)
    data_fisik_activation_1.forward(data_fisik_dense_1.output)

    data_fisik_dense_2.set_parameters(weights=loaded_model["data_fisik_dense_2"][0], biases=loaded_model["data_fisik_dense_2"][1])
    data_fisik_dense_2.forward(data_fisik_batch1.forward(data_fisik_activation_1.output))
    data_fisik_activation_2.forward(data_fisik_dense_2.output)
    combination_value = np.concatenate((R4_batch.forward(R4_activation.output), \
                                       G4_batch.forward(G4_activation.output), \
                                       B4_batch.forward(B4_activation.output), \
                                       data_fisik_batch2.forward(data_fisik_activation_2.output)), axis=1)

    dense_1.set_parameters(weights=loaded_model["dense_1"][0], biases=loaded_model["dense_1"][1])
    dense_1.forward(combination_value)
    dense1_activation.forward(dense_1.output)

    dense_2.set_parameters(weights=loaded_model["dense_2"][0], biases=loaded_model["dense_2"][1])
    dense_2.forward(dense1_activation.output)
    dense2_activation.forward(dense_2.output)

    dense_3.set_parameters(weights=loaded_model["dense_3"][0], biases=loaded_model["dense_3"][1])
    dense_3.forward(dense2_activation.output)
    loss_activation.forward(dense_3.output)

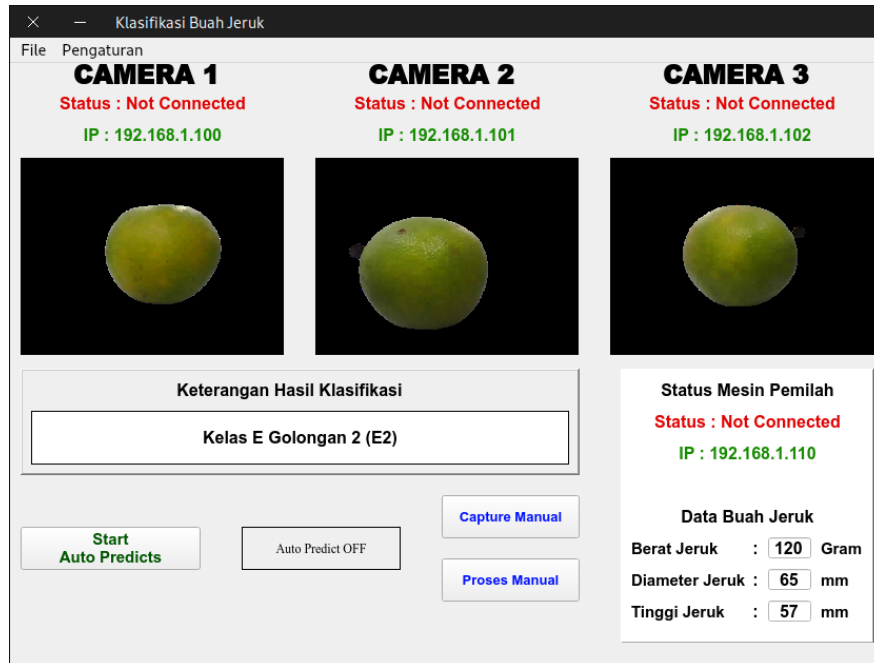
    predicts = np.argmax(loss_activation.output, axis=1)
    # np.set_printoptions(threshold=np.inf)
    # print(fisik_test[-1])
    # print(R_test[0].shape)

    predicts = predicts[0]

```

Data yang telah dinormalisasi kemudian dimasukkan ke dalam perintah pemanggilan model untuk dihitung nilai probabilitas tiap kelas. Hasil Prediksi nilai probabilitas akan dibaca nilai maksimum untuk memperoleh nilai kelas dari

data masukan. Pada gambar 3.13 menunjukkan hasil masukan data yang telah melalui beberapa proses sebelumnya dan dikategorikan ke dalam kelas E2.



Gambar 3. 13 Hasil Output

BAB IV

UJI COBA DAN PEMBAHASAN

4.1 Data Penelitian

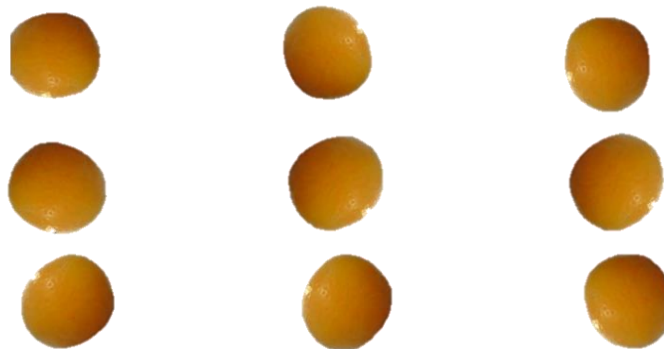
Data penelitian yang berhasil dikumpulkan dari tiap kelas kematangan A-E masing masing berjumlah A (25), B (25), C (25), D (25), dan E (30) buah jeruk. Total terdapat 130 data buah jeruk yang terdiri dari 3 gambar sisi buah jeruk dan data numerik yang mana merupakan data yang berhasil peneliti kumpulkan untuk penelitian ini. Data yang digunakan sudah dilakukan data *cleaning* terlebih dahulu untuk memastikan semua data terkategori dalam kelas yang sesuai.

Data yang diperoleh sebelumnya dilakukan *split data training* dan *testing* terlebih dahulu dengan perbandingan 70:30, dan 80:20, namun pada sistem ini tidak dilakukan pengujian perbandingan data 90:10 dikarenakan jumlah data *testing* setelah dilakukan *split* tidak mencukupi untuk keseluruhan kelas. Pada data training dilakukan augmentasi sebanyak 10 kali dengan tujuan agar data untuk proses training pada masing masing kelas memiliki data yang cukup memadai. Pada data 70:30 terdapat 91 data training original dan diaugmentasi 10x menjadi 910 data training dimana masing masing kelas memperoleh 45 data. Terakhir pada data 80:20 terdapat 104 data training original dan diaugmentasi 10x menjadi 1040 data training dimana masing masing kelas memperoleh 52 data.

Proses augmentasi gambar dilakukan sebagaimana pada Gambar 4.1 dari data kelas A yang diaugmentasi sebanyak 10 kali yang hasilnya terdapat pada Gambar 4.2.



Gambar 4. 1 Contoh data gambar asli jeruk kelas kematangan A



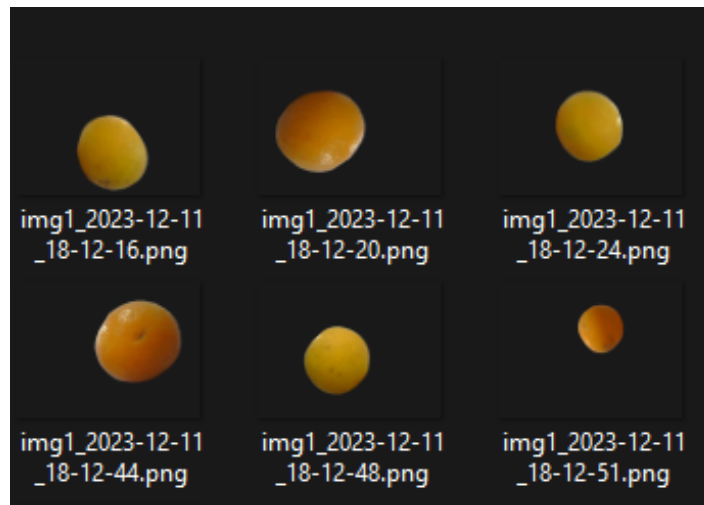
Gambar 4. 2 Contoh hasil augmentasi data dari kelas kematangan A

Data dimuat dalam bentuk format csv yang terdiri dari id nama gambar sisi 1-3, ukuran objek (berat, tinggi, diameter), dan klasifikasi kelas.

	id_gambar1	id_gambar2	id_gambar3	berat	tinggi	diameter	klasifikasi
1	img1_2023-12-11_18-12-16.png	img2_2023-12-11_18-12-18.png	img3_2023-12-11_18-12-19.png	56	41	47	A4
2	img1_2023-12-11_18-12-20.png	img2_2023-12-11_18-12-22.png	img3_2023-12-11_18-12-23.png	70	41	47	A4
3	img1_2023-12-11_18-12-24.png	img2_2023-12-11_18-12-26.png	img3_2023-12-11_18-12-27.png	52	43	50	A4
4	img1_2023-12-11_18-12-28.png	img2_2023-12-11_18-12-29.png	img3_2023-12-11_18-12-31.png	64	33	41	A4
5	img1_2023-12-11_18-12-32.png	img2_2023-12-11_18-12-33.png	img3_2023-12-11_18-12-35.png	57	36	48	A4
6	img1_2023-12-11_18-12-36.png	img2_2023-12-11_18-12-37.png	img3_2023-12-11_18-12-39.png	74	34	48	A4
7	img1_2023-12-11_18-12-40.png	img2_2023-12-11_18-12-41.png	img3_2023-12-11_18-12-42.png	54	33	46	A4
8	img1_2023-12-11_18-12-44.png	img2_2023-12-11_18-12-45.png	img3_2023-12-11_18-12-46.png	70	39	50	A4
9	img1_2023-12-11_18-12-48.png	img2_2023-12-11_18-12-49.png	img3_2023-12-11_18-12-50.png	76	42	48	A4
10							

Gambar 4. 3 Contoh data yang tersimpan dalam format .csv

Data format csv berfungsi untuk memanggil data gambar yang telah dilakukan proses *remove background* terlebih dahulu, dimana penamaan file gambar sama dengan nama yang tersimpan pada file format csv. Gambar 4.3 merupakan contoh penamaan data gambar yang digunakan.



Gambar 4. 4 Data gambar yang digunakan untuk proses training

4.2 Skenario Uji Coba

Skenario uji coba yang dilakukan untuk menentukan jumlah *Hidden neuron* didasarkan pada penelitian Shibata & Ikeda (2009) berjudul “Effect of Number of *Hidden Neurons on Learning in Large-Scale Layered Neural Networks*” dengan sedikit penyesuaian pada beberapa sistem. Penelitian tersebut membahas terkait panduan untuk menentukan jumlah *Hidden neuron* pada algoritma *Neural Network* dengan arsitektur yang cukup kompleks sesuai dengan sifat kompleksitas data yang digunakan pada penelitian ini. Persamaan untuk menghitung jumlah tiap *neuron* yaitu,

$$Z^{(h)} = \sqrt{Z^{(i)} Z^{(o)}} \quad (4.1)$$

Keterangan :

- $Z^{(h)}$ = Jumlah *Hidden neuron*
- $Z^{(i)}$ = Jumlah parameter *input*
- $Z^{(o)}$ = Jumlah output kelas

Skenario uji coba yang dibuat menggunakan 3 arsitektur yang berbeda dimana dibagi menjadi sistem 1 (*lower complexcity* model) yang menggunakan single *Layer* untuk setiap parameter data yang berbeda, sistem 2 (*medium complexcity* model) menerapkan arsitektur dengan Multi *Layer* Perceptron, dan sistem 3 (*most complexcity* model) dengan menambahkan jumlah *Hidden Layer* pada sistem 2 untuk mengetahui apakah model yang semakin kompleks memberikan pengaruh yang cukup signifikan atau tidak terhadap tingkat akurasi.

4.2.1 Perhitungan Model Sistem 1

Perhitungan jumlah *Hidden neuron* pada sistem 1 pada parameter data *image* dengan ukuran dimensi 200x600 dimana nilai *RGB* sudah dipisah untuk menjadi *input Layer* berbeda dan data numerik dengan 3 *input* parameter sebagai berikut,

$$\begin{aligned} Z(h_{image}) &= \sqrt{Z^{(i)} Z^{(o)}} \\ &= \sqrt{(200 \times 600) \times (20)} \\ &= 1549.193 \approx 1549 \end{aligned}$$

$$\begin{aligned} Z(h_{numerik}) &= \sqrt{Z^{(i)} Z^{(o)}} \\ &= \sqrt{(3) \times (20)} \\ &= 7.745 \approx 8 \end{aligned}$$

Tabel 4. 1 Model *Neural Network* Sistem 1

Nama	Detail Arsitektur
Sistem 1	<ul style="list-style-type: none"> - <i>Input Layer</i> (<i>Red Color, Green Color, Blue Color, Size Object</i>) - <i>Hidden Layer Red</i> : 1x (1549) dan <i>Batch Normalization</i> - <i>Hidden Layer Green</i> : 1x (1549) dan <i>Batch Normalization</i> - <i>Hidden Layer Blue</i> : 1x (1549) dan <i>Batch Normalization</i> - <i>Hidden Layer Size</i> : 1x (8) dan <i>Batch Normalization</i> - <i>Hidden Layer after concantenate</i> : 1x (20) - <i>Output Layer</i>

4.2.2 Perhitungan Model Sistem 2

Perhitungan jumlah *Hidden neuron* pada sistem 2 memiliki sedikit perubahan pada persamaan yang digunakan dimana *Hidden Layer* pertama yang terhubung dengan *input Layer* dibagi dengan dua, hal ini untuk menghindari sistem yang terlalu berat sehingga gagal diproses oleh komputer karena terdapat penambahan *Hidden layer*. Untuk jumlah *Hidden neuron* selanjutnya adalah hasil bagi dua jumlah *Hidden neuron* sebelumnya.

$$\begin{aligned}
 Z^{(h1_{image})} &= \sqrt{Z^{(i)} Z^{(o)}} \\
 &= \frac{\sqrt{(200 \times 600) \times (20)}}{2} \\
 &= 774.596 \approx 775 \\
 Z^{(h2_{image})} &= \frac{Z^{(h1_{image})}}{2} \\
 &= \frac{774.596}{2} = 387.298 \approx 387
 \end{aligned}$$

Karena terdapat perbedaan jumlah *neuron* yang terlalu tinggi antara *Hidden Layer image* dan *Hidden Layer numeric* pada sistem 1, maka hasil perhitungan *Hidden Layer numeric* ke-1 pada sistem 2 dikalikan dengan 6 untuk mengetahui pengaruh jumlah *neuron* setiap *Hidden Layer* dari *input* berbeda apakah harus sejajar atau proporsional. Dan pada *Hidden Layer numeric* ke-2 adalah hasil kali 2 *Hidden Layer numeric* ke-1.

$$\begin{aligned}
 Z^{(h_{numerik1})} &= \sqrt{Z^{(i)} Z^{(o)}} \\
 &= \sqrt{(3) \times (20)} \\
 &= 7.745 \approx 8 \times 6 = 48
 \end{aligned}$$

$$\begin{aligned}
 Z^{(h_{\text{numerik2}})} &= \frac{Z^{(h_{\text{numerik1}})}}{2} \\
 &= \frac{48}{2} = 24
 \end{aligned}$$

Jumlah *Hidden neuron* pertama setelah concatenate *Layer* ditentukan dengan jumlah *Hidden neuron* ke-2 data *image* R, G, B, dan *Hidden neuron* ke-2 data numerik.

$$\begin{aligned}
 Z^{(h1)} &= \sqrt{Z^{(h2_{\text{red}})} Z^{(h2_{\text{red}})} Z^{(h2_{\text{red}})} Z^{(h_{\text{numerik2}})}} \\
 &= \sqrt{(387 + 387 + 387 + 24) \times (20)} = 153.948 \approx 154 \\
 Z^{(h2)} &= \frac{Z^{(h1)}}{2} = \frac{154}{2} = 77
 \end{aligned}$$

Tabel 4. 2 Model *Neural Network* Sistem 2

Nama	Detail Arsitektur
Sistem 2	<ul style="list-style-type: none"> - <i>Input Layer</i> (Red Color, Green Color, Blue Color, Size Object) - <i>Hidden Layer Red</i> : 2x (775, 387) dan <i>Batch Normalization</i> - <i>Hidden Layer Green</i> : 2x (775, 387) dan <i>Batch Normalization</i> - <i>Hidden Layer Blue</i> : 2x (775, 387) dan <i>Batch Normalization</i> - <i>Hidden Layer Size</i> : 2x (48, 24) dan <i>Batch Normalization</i> - <i>Hidden Layer after concatenate</i> : 3x (154, 77, 20) - <i>Output Layer</i>

4.2.3 Perhitungan Model Sistem 3

Arsitektur sistem 3 yang digunakan terdapat penambahan 2 *Hidden Layer* disetiap *Layer* yang mengolah data *image* dan jumlah *neuron Hidden Layer* ke-2 data fisik adalah hasil kali 2 dari *Hidden Layer* ke-1 data fisik agar memiliki jumlah *Hidden Layer* akhir sebelum concatenate *Layer* yang hampir sama dengan *Hidden Layer* ke-4 data gambar untuk mengetahui efek persebaran *neuron* yang merata memiliki pengaruh klasifikasi kelas ukuran atau tidak. Penambahan *Layer* dan perubahan jumlah *neuron* dilakukan untuk mengetahui pengaruh

kompleksitas model untuk mengolah tipe data yang kompleks akan memberikan pengaruh yang cukup signifikan atau tidak. Jumlah *Hidden neuron* pertama setelah concatenate *Layer* ditentukan dengan jumlah *Hidden neuron* ke-4 data *image* R, G, B, dan *Hidden neuron* ke-2 data numerik.

$$\begin{aligned}
 Z^{(h1_{image})} &= \sqrt{Z^{(i)} Z^{(o)}} \\
 &= \frac{\sqrt{(200 \times 600) \times (20)}}{2} \\
 &= 774.596 \approx 775 \\
 Z^{(h2_{image})} &= \frac{Z^{(h1_{image})}}{2} \\
 &= \frac{774.596}{2} = 387.298 \approx 387 \\
 Z^{(h3_{image})} &= \frac{Z^{(h2_{image})}}{2} \\
 &= \frac{387}{2} = 193.5 \approx 194 \\
 Z^{(h4_{image})} &= \frac{Z^{(h3_{image})}}{2} \\
 &= \frac{194}{2} = 97 \\
 Z^{(h_{numerik1})} &= \sqrt{Z^{(i)} Z^{(o)}} \\
 &= \sqrt{(3) \times (20)} \\
 &= 7.745 \approx 8 \times 6 = 48 \\
 Z^{(h_{numerik2})} &= \frac{Z^{(h_{numerik1})}}{2} \\
 &= 48 \times 2 = 96
 \end{aligned}$$

$$\begin{aligned}
 Z^{(h1)} &= \sqrt{Z^{(h4_{red})} Z^{(h4_{red})} Z^{(h4_{red})} Z^{(h_{numerik2})}} \\
 &= \sqrt{(97 + 97 + 97 + 96) \times (20)} = 87.977 \approx 88 \\
 Z^{(h2)} &= \frac{N^{(h1)}}{2} = \frac{88}{2} = 44
 \end{aligned}$$

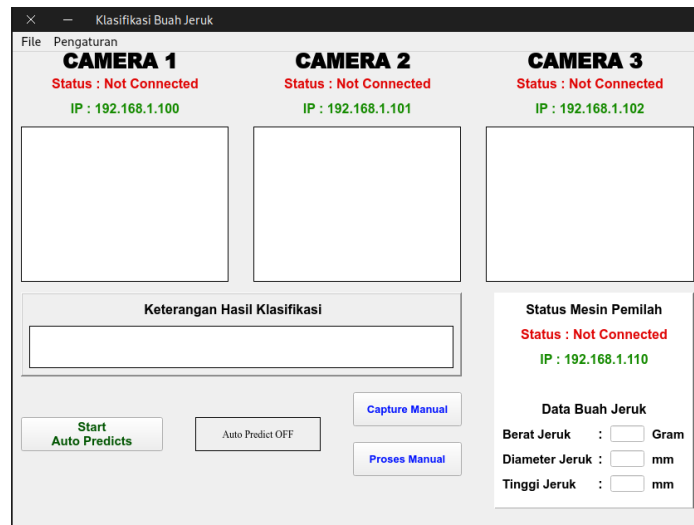
Tabel 4. 3 Model *Neural Network* Sistem 3

Nama	Detail Arsitektur
Sistem 3	<ul style="list-style-type: none"> - <i>Input Layer</i> (<i>Red Color, Green Color, Blue Color, Size Object</i>) - <i>Hidden Layer Red</i> : 4x (775, 387, 194, 97) dan <i>Batch Normalization</i> - <i>Hidden Layer Green</i> : 4x (775, 387, 194, 97) dan <i>Batch Normalization</i> - <i>Hidden Layer Blue</i> : 4x (775, 387, 194, 97) dan <i>Batch Normalization</i> - <i>Hidden Layer Size</i> : 2x (48, 96) dan <i>Batch Normalization</i> - <i>Hidden Layer after concantenate</i> : 3x (88, 44, 20) - <i>Output Layer</i>

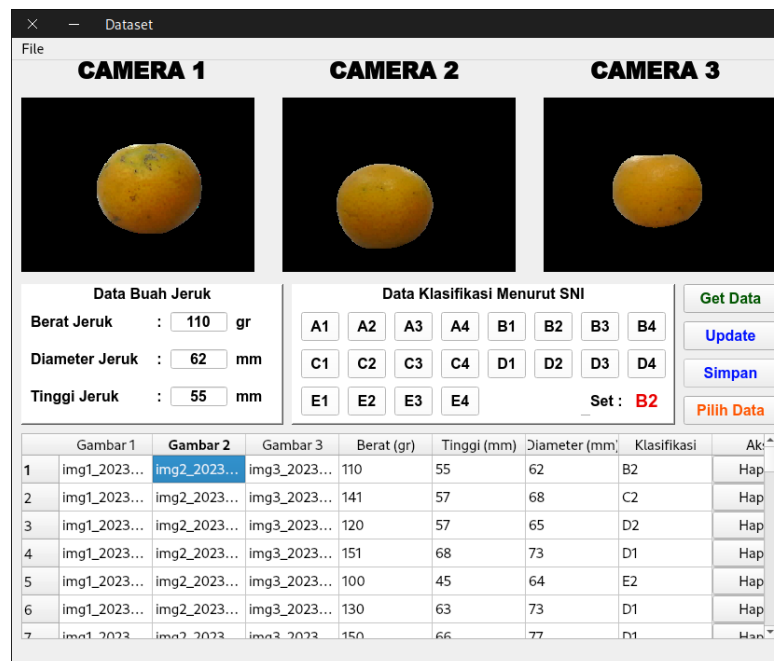
Setelah membuat tiga model berbeda maka ditentukan nilai *learning rate* *epoch alpha* (0,001), lama menunggu jika tidak ada perbaikan nilai *loss* bernilai 5 perulangan, dan *batch size* (225)

4.3 Aplikasi Desktop dan Hardware

Hasil dari model yang telah *training* menggunakan *Neural Network* dari setiap sistem disimpan dalam bentuk format .pkl. Model tersebut berisi nilai bobot dan bias dari setiap model yang kemudian disimpan pada program desktop yang telah dibuat agar memiliki *knowledge* untuk melakukan pengklasifikasian Ketika ada masukan baru. Gambar 4.3 menunjukkan interface utama yang nantinya program desktop akan terhubung dengan hardware untuk membaca masukan baru berupa data gambar 3 sisi dan ukuran dari objek. Gambar 4.4 menunjukkan halaman kedua untuk melakukan pengujian dengan data yang telah tersimpan pada file .csv ketika aplikasi desktop tidak terhubung dengan hardware.



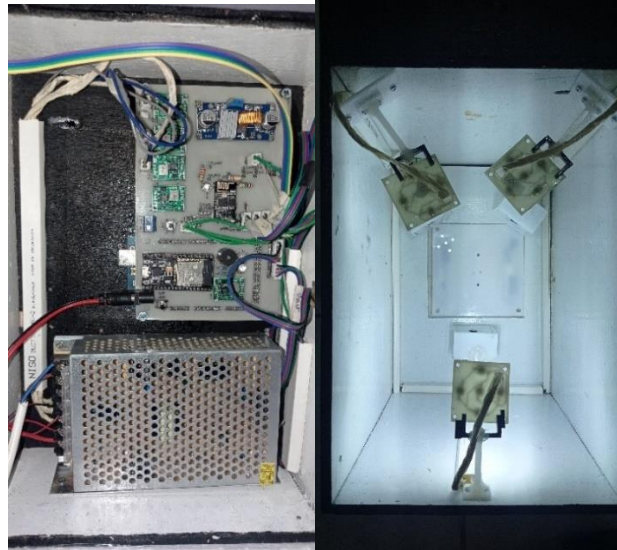
Gambar 4. 5 Interface utama program



Gambar 4. 6 Interface pengujian data yang tersimpan

Antara software dan hardware terjadi komunikasi dua arah menggunakan protokol http. Kedua perangkat yang telah terpasang software yang telah dibuat berada satu jaringan pada hardware agar bisa berkomunikasi secara lokal.

Perangkat hardware terdiri dari 3 esp32cam, sensor lidar, sensor load cell, power management, Arduino mega yang terpasang esp01, dan esp32.



Gambar 4. 7 Hardware tampak dalam dan komponen

4.4 Menghitung Kinerja Sistem

Kinerja sistem yang telah dibangun akan dihitung performanya menggunakan *confusion matrix*. Pada setiap sistem uji coba dilakukan perhitungan terkait nilai recall, tingkat akurasi, dan presisi, *recall*, dan pengukuran *f-measure* untuk mengetahui performa pada masing masing sistem uji coba. Proses pengukuran dilakukan menggunakan *confusion matrix* multiclass. Perhitungan dilakukan dengan membandingkan kelas sebenarnya dengan kelas hasil prediksi. Nilai *confusion matrix* terdiri dari *True Positive* (TP), *False Positive* (FP), *True Negative* (TN), dan *False Negative* (FN). Masing masing memiliki makna sebagai berikut:

A. *True Positive* : Nilai yang diprediksi benar menunjukkan data yang benar.

B. *False Negative* : Nilai yang diprediksi negatif merupakan nilai yang sebenarnya negatif.

C. *False Positive* : Jumlah data yang diprediksi negative merupakan nilai yang sebenarnya positif.

D. *True Negative*: Banyaknya nilai negatif yang diprediksi terdeteksi benar.

Pencarian nilai TP, FN, FP, dan TN berfungsi untuk menghitung nilai dari akurasi, presisi, *recall*, dan *f-measure*. Analisa dilakukan pada setiap kelas yang dihasilkan dari arsitektur percobaan yang digunakan untuk mengetahui kemampuan tiap arsitektur dalam mengklasifikasikan suatu kelas, dengan mengetahui kemampuan klasifikasi suatu kelas maka dapat disimpulkan faktor yang mempengaruhi hasil kinerja dari suatu arsitektur dari korelasi nilai kinerja sistem, data yang digunakan, dan arsitektur yang dibuat. Untuk perhitungan nilai akurasi, presisi, *recall*, dan *f-measure* berdasarkan jurnal yang ditulis Grandini et al., (2020) berjudul “*Metrics for Multi-Class Classification: An Overview*” yang membahas pengujian performa indikator pada model yang dapat mengklasifikasikan lebih dari 2 kelas, dimana persamaan tersebut sebagai berikut:

A. *Accuracy* adalah rasio Prediksi benar dari keseluruhan data.

$$\text{Balanced Accuracy} = \frac{\sum_i \frac{TP[i]}{\text{TotalRow}[i]}}{\text{Jumlah_kelas}} \times 100 \% \quad (4.2)$$

B. *Precision* adalah rasio Prediksi *True* Positif dibandingkan dengan keseluruhan hasil yang diPrediksi positif.

$$\text{Precision}_c = \frac{TP_c}{TP_c + FP_c} \times 100\% \quad (4.3)$$

C. *Recall* adalah rasio Prediksi *True* Positif dibandingkan dengan keseluruhan data yang benar positif.

$$Recall = \frac{TP_c}{TP_c + FN_c} \times 100\% \quad (4.4)$$

D. *F-measure* adalah keseluruhan perbandingan rata rata presisi dan *recall* yang dibobotkan.

$$F - measure = \frac{2 \times Precision_c \times Recall_c}{Precision_c + Recall_c} \quad (4.5)$$

4.5 Hasil Uji Coba

Kinerja yang akan diuji menggunakan *confusion matrix* dari setiap sistem adalah kombinasi data terbaik dari masing masing sistem yang telah dibuat dengan tiap data testing yang digunakan sebelumnya. Hasil Prediksi sistem akan dibandingkan dengan data aktual, apakah sistem mampu mengklasifikasikan jawaban dengan benar atau tidak.

Tabel 4.4 merupakan hasil yang diperoleh ketika tahap training uji coba setiap sistem dengan hasil *accuracy* dan *validation accuracy* yang beragam. Pada sistem 1 (*lower complexity*) tidak mampu mengidentifikasi data dengan baik dimana akurasi terbaik pada sistem 1 70:30 dengan akurasi 37.8% dan validasi akurasi 26.9%. Pada sistem 2 (*medium complexity*) menghasilkan model yang jauh membaik dari sistem 1, namun belum cukup menghasilkan model yang mampu mengklasifikasikan data dengan akurasi tinggi dengan hasil terbaik pada sistem 2 (80:20) yang memiliki akurasi 68.3% dan validasi akurasi 60.9%.. Hasil sistem 3 (*most complexity*) memiliki model dengan akurasi yang cukup baik

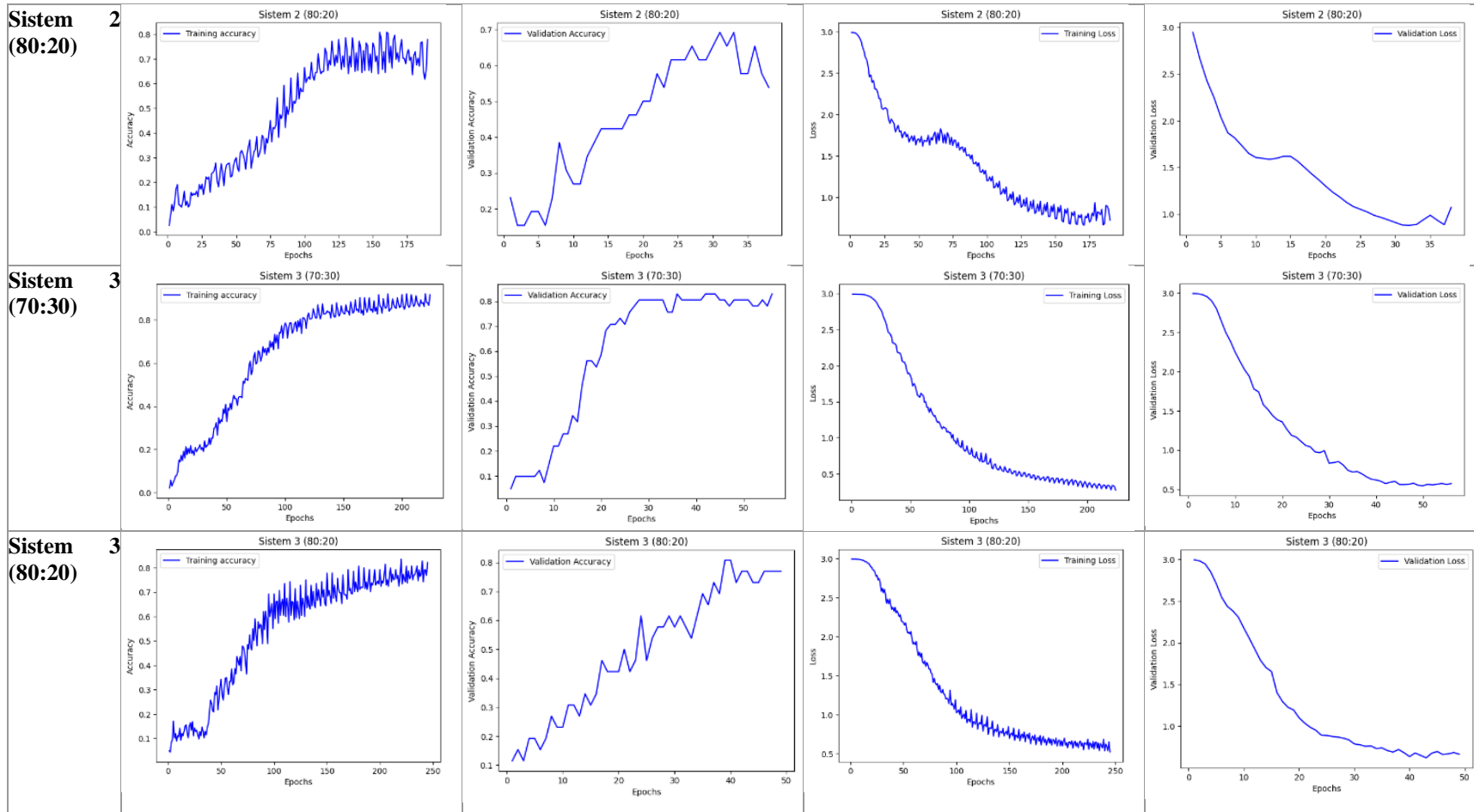
dengan rata rata akurasi lebih dari 70% dengan nilai terbaik pada sistem 3 (80:20) yang memiliki nilai *accuracy* tertinggi dengan selisih *accuracy* dan *validation accuracy* terendah dengan hasil akurasi 77.1% dan validasi akurasi 80.8%.

Tabel 4. 4 Hasil uji coba setiap sistem

Sistem	<i>Accuracy</i>	<i>Loss</i>	<i>Validation Accuracy</i>	<i>Validation Loss</i>
Sistem 1 (70:30)	32.1 %	1.997	17.1 %	2.111
Sistem 1 (80:20)	37.8%	1.846	26.9%	1.810
Sistem 2 (70:30)	68.3%	1.062	60.9%	1.397
Sistem 2 (80:20)	77.8 %	0.725	53.8 %	1.074
Sistem 3 (70:30)	83.1 %	0.429	70.7 %	0.710
Sistem 3 (80:20)	77.1 %	0.571	80.8 %	0.456

Tabel 4. 5 Grafik hasil uji coba setiap sistem

Sistem	Accuracy	Validation Accuracy	Loss	Validation Loss
Sistem 1 (70:30)				
Sistem 1 (80:20)				
Sistem 2 (70:30)				



4.5.1 Sistem 1 (70:30)

Tabel 4. 6 Perhitungan *Confusion Matrix* Sistem 1 (70:30)

Hasil Prediksi				Confusion Matrix						
Y True	Y Pred	Y True	Y Pred	Kelas	FP	FN	TN	TP	Total Kelas	<u>TP</u> Total row
A1	B2	C3	B2	A1	1	4	36	0	1	0,00
A2	A2	C4	B3	A2	1	2	37	1	2	0,50
A2	B1	C4	A3	A3	2	3	35	1	3	0,33
A3	A3	D1	D2	A4	1	0	39	1	2	0,50
A3	A1	D2	D2	B1	1	2	38	0	1	0,00
A3	A1	D2	E1	B2	2	3	36	0	2	0,00
A4	A4	D3	E4	B3	3	3	35	0	3	0,00
A4	B3	D3	E4	B4	2	0	39	0	2	0,00
B1	A1	D3	D4	C1	1	1	39	0	1	0,00
B2	B1	D4	E2	C2	2	1	38	0	2	0,00
B2	A1	D4	E3	C3	3	0	38	0	3	0,00
B3	A2	E1	E2	C4	2	0	39	0	2	0,00
B3	A3	E1	E1	D1	1	0	40	0	1	0,00
B3	B2	E2	E1	D2	1	1	38	1	2	0,50
B4	A3	E2	E1	D3	3	1	37	0	3	0,00
B4	B3	E3	E1	D4	2	2	37	0	2	0,00
C1	C2	E3	E2	E1	1	4	35	1	2	0,50
C2	A2	E3	E4	E2	2	3	36	0	2	0,00
C2	C1	E4	E4	E3	3	2	36	0	3	0,00
C3	D3	E4	E3	E4	1	3	36	1	2	0,50
C3	D4			Total	35	35	744	6	41	2,83

$$a. Accuracy = \frac{\sum_i \frac{TP[i]}{Total_row[i]}}{Jumlah_kelas} \times 100 \% = \frac{2,83}{20} \times 100 \% = 14.2 \%$$

$$b. Precision_c = \frac{TP_c}{TP_c + FP_c} \times 100 \% = \frac{6}{6 + 35} \times 100 \% = 14.6 \% \%$$

$$c. Recall = \frac{TP_c}{TP_c + FN_c} \times 100 \% = \frac{6}{6 + 35} \times 100 \% = 14.6\%$$

$$d. F\ measure = \frac{2 \times Precision_c \times Recall_c}{Precision_c + Recall_c} = \frac{2 \times 14.6 \times 14.6}{14.6 + 14.6} = 14.6$$

4.5.2 Sistem 1 (80:20)

Tabel 4. 7 Perhitungan *Confusion Matrix* Sistem 1 (80:20)

Hasil Prediksi		Confussion Matrix						
Y True	Y Pred	Kelas	FP	FN	TN	TP	Total Kelas	$\frac{TP}{Total\ row}$
A1	A1	A1	0	1	24	1	1	1,00
A2	A1	A2	1	0	25	0	1	0,00
A3	A4	A3	1	2	22	1	2	0,50
A3	A3	A4	1	1	24	0	1	0,00
A4	A3	B1	0	4	21	1	1	1,00
B1	B1	B2	1	0	25	0	1	0,00
B2	B1	B3	1	1	23	1	2	0,50
B3	B4	B4	1	1	24	0	1	0,00
B3	B3	C1	1	0	25	0	1	0,00
B4	B3	C2	1	0	25	0	1	0,00
C1	B1	C3	2	0	24	0	2	0,00
C2	B1	C4	1	0	25	0	1	0,00
C3	B1	D1	1	0	25	0	1	0,00
C3	D4	D2	0	2	23	1	1	1,00
C4	A3	D3	2	1	23	0	2	0,00
D1	D2	D4	1	2	23	0	1	0,00
D2	D2	E1	1	2	22	1	2	0,50
D3	D4	E2	1	1	24	0	1	0,00
D3	D2	E3	1	1	23	1	2	0,50
D4	D3	E4	1	0	25	0	1	0,00
E1	E1	Total	19	19	475	7	26	5,00
E1	E2							
E2	E1							
E3	E1							
E3	E3							
E4	E3							

$$a. Accuracy = \frac{\sum_i \frac{TP[i]}{Total_row[i]}}{Jumlah_kelas} \times 100\% = \frac{5,00}{20} \times 100\% = 25,0\%$$

$$b. Precision_c = \frac{TP_c}{TP_c + FP_c} \times 100\% = \frac{7}{7 + 19} \times 100\% = 26,9\%$$

$$c. Recall = \frac{TP_c}{TP_c + FN_c} \times 100\% = \frac{7}{7 + 19} \times 100\% = 26,9\%$$

$$d. F\ measure = \frac{2 \times Precision_c \times Recall_c}{Precision_c + Recall_c} = \frac{2 \times 26,9 \times 26,9}{26,9 + 26,9} = 26,9$$

4.5.3 Sistem 2 (70:30)

Tabel 4. 8 Perhitungan *Confusion Matrix* Sistem 2 (70:30)

Hasil Prediksi				<i>Confusion Matrix</i>						
Y True	Y Pred	Y True	Y Pred	Kelas	FP	FN	TN	TP	Total Kelas	$\frac{TP}{Total\ row}$
A1	A1	C3	A3	A1	0	1	39	1	1	1,00
A2	A2	C4	A4	A2	0	1	38	2	2	1,00
A2	A2	C4	D4	A3	1	2	36	2	3	0,67
A3	A3	D1	D1	A4	2	1	38	0	2	0,00
A3	C3	D2	D1	B1	0	0	40	1	1	1,00
A3	A3	D2	D2	B2	1	1	38	1	2	0,50
A4	A3	D3	D3	B3	1	0	38	2	3	0,67
A4	B4	D3	D3	B4	1	1	38	1	2	0,50
B1	B1	D3	D3	C1	1	0	40	0	1	0,00
B2	A2	D4	D4	C2	1	0	39	1	2	0,50
B2	B2	D4	C4	C3	2	1	37	1	3	0,33
B3	B3	E1	E1	C4	2	1	38	0	2	0,00
B3	B3	E1	E1	D1	0	1	39	1	1	1,00
B3	D3	E2	E1	D2	1	0	39	1	2	0,50
B4	B4	E2	E2	D3	0	3	35	3	3	1,00
B4	D4	E3	E3	D4	1	2	37	1	2	0,50
C1	A1	E3	D3	E1	0	1	38	2	2	1,00
C2	B2	E3	E3	E2	1	0	39	1	2	0,50
C2	C2	E4	E4	E3	1	0	38	2	3	0,67
C3	C3	E4	E4	E4	0	0	39	2	2	1,00
C3	D3			Total	16	16	763	25	41	12,33

$$a. Accuracy = \frac{\sum_i \frac{TP[i]}{Total_row[i]}}{jumlah_kelas} \times 100\% = \frac{12,33}{20} \times 100\% = 61,65\%$$

$$b. Precision_c = \frac{TP_c}{TP_c + FP_c} \times 100\% = \frac{25}{25 + 16} \times 100\% = 60,9\%$$

$$c. Recall = \frac{TP_c}{TP_c + FN_c} \times 100\% = \frac{25}{25 + 16} \times 100\% = 60,9\%$$

$$d. F\ measure = \frac{2 \times Precision_c \times Recall_c}{Precision_c + Recall_c} = \frac{2 \times 60,9\% \times 60,9\%}{60,9\% + 60,9\%} = 60,9\%$$

4.5.4 Sistem 2 (80:20)

Tabel 4. 9 Perhitungan *Confusion Matrix* Sistem 2 (80:20)

Hasil Prediksi		Confussion Matrix						
Y True	Y Pred	Kelas	FP	FN	TN	TP	Total Kelas	$\frac{TP}{Total\ row}$
A1	A1	A1	0	0	25	1	1	1,00
A2	B2	A2	1	0	25	0	1	0,00
A3	A3	A3	1	0	24	1	2	0,50
A3	B3	A4	1	0	25	0	1	0,00
A4	B4	B1	0	1	24	1	1	1,00
B1	B1	B2	0	2	23	1	1	1,00
B2	B2	B3	0	3	21	2	2	1,00
B3	B3	B4	1	1	24	0	1	0,00
B3	B3	C1	1	0	25	0	1	0,00
B4	B3	C2	1	0	25	0	1	0,00
C1	B1	C3	1	1	23	1	2	0,50
C2	B2	C4	0	1	24	1	1	1,00
C3	B3	D1	0	1	24	1	1	1,00
C3	C3	D2	0	0	25	1	1	1,00
C4	C4	D3	1	0	24	1	2	0,50
D1	D1	D4	1	0	25	0	1	0,00
D2	D2	E1	2	0	24	0	2	0,00
D3	C3	E2	0	2	23	1	1	1,00
D3	D3	E3	1	0	24	1	2	0,50
D4	C4	E4	0	0	25	1	1	1,00
E1	D1	Total	12	12	482	14	26	11,00
E1	E2							
E2	E2							
E3	E3							
E3	E2							
E4	E4							

$$a. \text{ Accuracy} = \frac{\sum_i \frac{TP[i]}{Total_row[i]}}{Jumlah_kelas} \times 100 = \frac{11,00}{20} \times 100 = 55,0 \%$$

$$b. \text{ Precision}_c = \frac{TP_c}{TP_c + FP_c} \times 100 = \frac{14}{14 + 12} \times 100 = 53,8 \%$$

$$c. \text{ Recall} = \frac{TP_c}{TP_c + FN_c} \times 100 = \frac{14}{14 + 12} \times 100 = 53,8 \%$$

$$d. \text{ F measure} = \frac{2 \times \text{Precision}_c \times \text{Recall}_c}{\text{Precision}_c + \text{Recall}_c} \times 100 = \frac{2 \times 53,8 \times 53,8}{53,8 + 53,8} = 53,8$$

4.5.5 Sistem 3 (70:30)

Tabel 4. 10 Perhitungan *Confusion Matrix* Sistem 3 (70:30)

Hasil Prediksi				Confussion Matrix						
Y True	Y Pred	Y True	Y Pred	Kelas	FP	FN	TN	TP	Total Kelas	$\frac{TP}{Total\ row}$
A1	A1	C3	B3	A1	0	1	39	1	1	1,00
A2	A2	C4	B4	A2	1	0	39	1	2	0,50
A2	A1	C4	C4	A3	0	1	37	3	3	1,00
A3	A3	D1	D1	A4	2	0	39	0	2	0,00
A3	A3	D2	D2	B1	0	1	39	1	1	1,00
A3	A3	D2	D2	B2	1	0	39	1	2	0,50
A4	A3	D3	D2	B3	1	2	36	2	3	0,67
A4	B4	D3	D3	B4	1	2	37	1	2	0,50
B1	B1	D3	D3	C1	0	0	40	1	1	1,00
B2	B1	D4	D4	C2	0	1	38	2	2	1,00
B2	B2	D4	D4	C3	3	1	37	0	3	0,00
B3	B3	E1	E1	C4	1	1	38	1	2	0,50
B3	C3	E1	E1	D1	0	0	40	1	1	1,00
B3	B3	E2	E1	D2	0	1	38	2	2	1,00
B4	B4	E2	E2	D3	1	0	38	2	3	0,67
B4	C4	E3	E3	D4	0	0	39	2	2	1,00
C1	C1	E3	E3	E1	0	1	38	2	2	1,00
C2	C2	E3	E3	E2	1	0	39	1	2	0,50
C2	C2	E4	E4	E3	0	0	38	3	3	1,00
C3	C2	E4	E4	E4	0	0	39	2	2	1,00
C3	B3			Total	12	12	767	29	41	14,83

$$a. Accuracy = \frac{\sum_i \frac{TP[i]}{Total_row[i]}}{jumlah_kelas} \times 100\% = \frac{14,83}{20} \times 100\% = 74,17\%$$

$$b. Precision_c = \frac{TP_c}{TP_c + FP_c} \times 100\% = \frac{29}{29+12} \times 100\% = 70,7\%$$

$$c. Recall = \frac{TP_c}{TP_c + FN_c} \times 100\% = \frac{29}{29+12} \times 100\% = 70,7\%$$

$$d. F\ measure = \frac{2 \times Precision_c \times Recall_c}{Precision_c + Recall_c} = \frac{2 \times 70,7 \times 70,7}{70,7 + 70,7} = 70,7$$

4.5.6 Sistem 3 (80:20)

Tabel 4. 11 Perhitungan *Confusion Matrix* Sistem 2 (80:20)

Hasil Prediksi		Confussion Matrix						
Y True	Y Pred	Kelas	FP	FN	TN	TP	Total Kelas	$\frac{TP}{Total\ row}$
A1	A1	A1	0	0	25	1	1	1,00
A2	A2	A2	0	0	25	1	1	1,00
A3	A3	A3	0	0	24	2	2	1,00
A3	A3	A4	1	0	25	0	1	0,00
A4	B4	B1	0	1	24	1	1	1,00
B1	B1	B2	0	0	25	1	1	1,00
B2	B2	B3	0	1	23	2	2	1,00
B3	B3	B4	0	2	23	1	1	1,00
B3	B3	C1	1	0	25	0	1	0,00
B4	B4	C2	0	0	25	1	1	1,00
C1	B1	C3	1	0	24	1	2	0,50
C2	C2	C4	1	0	25	0	1	0,00
C3	B3	D1	1	0	25	0	1	0,00
C3	C3	D2	0	0	25	1	1	1,00
C4	B4	D3	0	0	24	2	2	1,00
D1	E1	D4	0	0	25	1	1	1,00
D2	D2	E1	0	1	23	2	2	1,00
D3	D3	E2	0	0	25	1	1	1,00
D3	D3	E3	0	0	24	2	2	1,00
D4	D4	E4	0	0	25	1	1	1,00
E1	E1	Total	5	5	489	21	26	15,50
E1	E1							
E2	E2							
E3	E3							
E3	E3							
E4	E4							

$$a. \text{ Accuracy} = \frac{\sum_i \frac{TP[i]}{Total\ row[i]}}{jumlah\ kelas} \times 100 = \frac{15,50}{20} \times 100 = 77.5 \%$$

$$b. \text{ Precision}_c = \frac{TP_c}{TP_c + FP_c} \times 100 = \frac{21}{21+5} \times 100 = 80.8 \%$$

$$c. \text{ Recall} = \frac{TP_c}{TP_c + FN_c} \times 100 = \frac{21}{21+5} \times 100 = 80.8 \%$$

$$d. \text{ F measure} = \frac{2 \times \text{Precision}_c \times \text{Recall}_c}{\text{Precision}_c + \text{Recall}_c} = \frac{2 \times 80.8 \times 80.8}{80.83 + 80.8} = 80.8$$

4.6 Pembahasan

Melihat akurasi dari hasil uji validasi akurasi setiap model yang telah dibuat pada setiap arsitektur yang berbeda disimpulkan bahwa pada sistem 1 - 3 diperoleh hasil terbaik tiap sistem dimana sistem 1 dengan perbandingan data 80:20 menghasilkan *accuracy* 37.8 %, *loss* 1,846, *validation accuracy* 26.9%, *validation loss* 1.810. Sistem 2 dengan perbandingan data 70:30 menghasilkan *accuracy* 68.3%, *loss* 1.062, *validation accuracy* 60.9%, *validation loss* 1.397. Terakhir sistem 3 dengan perbandingan data 80:20 memiliki *accuracy* 77.1%, *loss* 0.571, *validation accuracy* 80.8%, *validation loss* 0.456. Performa tertinggi diperoleh sistem 3 dengan perbandingan data 80:20 dengan *accuracy* 77.1%, *loss* 0.571, *validation accuracy* 80.8%, *validation loss* 0.456.

Berdasarkan hasil uji coba yang telah dilakukan dengan menggunakan tiga arsitektur yang berbeda dibagi menjadi *lower complexity*, *medium complexity*, dan *most complexity* menghasilkan akurasi terbaik dengan arsitektur *most complexity* yang memiliki arsitektur paling kompleks dan persebaran jumlah *neuron* dari setiap *Hidden Layer* di parameter *input* yang paling merata sehingga data dapat diproses dengan baik. Hasil uji coba juga menunjukkan ketika menggunakan arsitektur *lower complexity* yang menggunakan *single Layer* untuk setiap parameter data *input* dengan total 5 *Hidden Layer* pada sistem 1 menghasilkan model yang cukup buruk dalam pengkelasan kualitas kematangan maupun kelas ukuran yang hanya menghasilkan perhitungan terbaik *accuracy* 37.8 %, *loss* 1,846, *validation accuracy* 26.9%, *validation loss* 1.810.

Ketika densitas ditingkatkan dengan menambahkan *Hidden Layer* di tiap parameter *input* maka terjadi peningkatan akurasi yang cukup signifikan serta model mampu mengenali kelas ukuran jauh lebih baik dibandingkan sistem 1 dengan sedikit kasus kesalahan dalam pengklasifikasian kelas ukuran sebagai contoh pada hasil sistem terbaik dengan *accuracy* 68.3%, *loss* 1.062, *validation accuracy* 60.9%, *validation loss* 1.397 kasus klasifikasi A4 yang teridentifikasi sebagai A3 sangat sedikit. Namun sistem ini masih belum cukup baik dalam mengenali kelas kematangan sebagai contoh kelas C3 teridentifikasi sebagai kelas B3 masih cukup banyak.

Pada sistem 3 yang ditingkatkan lagi densitasnya dengan menggunakan 4 *Hidden Layer* di tiap parameter *input* gambar dan dilakukan penambahan *neuron* pada *Hidden Layer* ke-2 data fisik agar memiliki jumlah yang hampir sama dengan *Hidden Layer* ke-4 data gambar menghasilkan model yang mampu mengklasifikasikan kelas kematangan dan ukuran lebih baik dibandingkan sistem 1 dan 2 yang menghasilkan sistem terbaik dengan hasil *accuracy* 77.1%, *loss* 0.571, *validation accuracy* 80.8%, *validation loss* 0.456. Kesalahan dalam pengklasifikasian kelas ukuran pada sistem 3 jarang terjadi dan hasil sistem 3 (80:20) tidak memiliki kesalahan dalam pengklasifikasian kelas ukuran. Namun Sistem 3 masih memiliki sejumlah kesalahan dalam pengklasifikasian kelas kematangan sehingga akurasi yang diperoleh tidak lebih dari 80%.

Sistem yang dibangun sudah memiliki hasil model yang cukup baik dengan akurasi yang mendekati 80%. Akurasi yang tinggi diperlukan oleh sistem agar mampu mengidentifikasi data baru kedalam kelas yang telah ditentukan dan

menghindari kesalahan kelasahan dalam identifikasi data. Hal ini serupa dengan apa yang dijelaskan dalam ayat Al-Quran. Allah *Subhanallahu wa ta'ala* berfirman dalam surah Al-Isra (17:35),:

وَأَوْفُوا الْكَيْلَ إِذَا كِلْتُمْ وَزِنُوا بِالْقِسْطَاسِ الْمُسْتَقِيمِ ذَلِكَ خَيْرٌ وَأَحْسَنُ تَأْوِيلًا

”Sempurnakanlah takaran apabila kamu menakar dan timbanglah dengan timbangan yang benar. Itulah yang paling baik dan paling bagus akibatnya.” (QS. Al-Isra : 35)

Dalam buku yang disusun oleh Ishaq, (2013) terjemahan Tafsir oleh Ibnu Katsir ayat tersebut menekankan secara eksplisit bahwa perhitungan dalam timbangan harus dilakukan secara akurat yaitu tidak miring, tidak melenceng, dan tidak kacau (bergetar). Dalam konteks penelitian ini dapat dikaitkan bahwa sistem yang dibuat diharuskan untuk disempurnakan agar memiliki tingkat akurasi yang baik, sehingga meminimalisir kesalahan dalam pengklasifikasian suatu data. Seperti yang dilakukan pada pengujian sistem 1 – 3 terdapat peningkatan akurasi dengan penyempurnaan arsitektur model yang digunakan, dimulai dari sistem 1 70:30 dengan akurasi 14.2% meningkat hasilnya pada sistem 3 80:20 dengan akurasi 77.5% yang mana dapat dikategorikan cukup baik. Namun masih diperlukan peningkatan lagi sesuai dengan anjuran QS. Al-Isra (17:35) sistem yang dibangun harus disempurnakan lagi agar menjadi lebih baik yang mana dijelaskan dalam ayat tersebut *”Itulah yang paling baik dan paling bagus akibatnya”*.

Selain itu sistem klasifikasi buah jeruk menggunakan metode *Neural Network* yang telah dibuat memiliki keunggulan karena dapat mengkategorikan data baru kedalam kelas atau kategori tertentu berdasarkan ciri unik setiap data

yang dimiliki. Dengan sedikit perbaikan dan peningkatan arsitektur model klasifikasi yang telah dibuat, sistem mampu mengurangi kesalahan manusia dalam pengklasifikasian yang disebabkan ketidak telitian, kelelahan, ataupun ketidaksempurnaan, serta penilaian yang subjektif oleh manusia. Sedangkan di dalam Al-Quran dengan tegas menekankan untuk berlaku adil dalam berbisnis mencakup kesesuaian kualitas produk yang ditawarkan dengan apa yang dijual sesuai harga yang diberikan. Allah *Subhanallahu wa ta'ala* berfirman dalam Qur'an Surah Al-Mutaffifin (83:1-3) :

وَيْلٌ لِّلْمُطَفِّفِينَ ﴿١﴾ الَّذِينَ إِذَا أَكْتَالُوا عَلَى النَّاسِ يَسْتَوْفُونَ ﴿٢﴾ وَإِذَا كَالُواهُمْ أَوْ وُزِنُوا لَهُمْ يُخْسِرُونَ ﴿٣﴾

“Celakalah bagi orang-orang yang curang (dalam menakar dan menimbang)! (yaitu) orang-orang yang apabila menerima takaran dari orang lain mereka minta dicukupkan, dan apabila mereka menakar atau menimbang (untuk orang lain), mereka mengurangi.” (QS. Al-Mutaffifin : 1-3)

Dalam buku yang disusun oleh Ishaq, (2013) terjemahan Tafsir oleh Ibnu Katsir yang menerangkan bahwa ayat tersebut menekankan untuk berlaku adil dalam jual beli yang secara eksplisit menjelaskan untuk tidak mengurangi timbangan dan takaran, baik dengan menambah jika minta timbangan dari orang lain ataupun mengurangi jika memberikan timbangan kepada mereka karena terdapat murka Allah didalamnya. Gambaran berlaku adil dalam jual beli yang digambarkan dengan menimbang secara sempurna, dapat diinterpretasikan sebagai suatu kewajiban dalam berdagang dimana produk yang dijual adalah produk yang sesuai terhadap kualitas yang ditawarkan. Dimana keterkaitan dalam penelitian ini untuk membantu tahap pasca panen dalam mengkategorikan jeruk dalam beberapa kelas yang berbeda sesuai ciri khusus yang dimiliki.

Manfaat atau kewajiban mengembangkan teknologi untuk kebaikan atau kepentingan manusia secara tidak langsung ditegaskan dalam HR. Muslim no. 1631 yang berisi,

إِذَا مَاتَ الْإِنْسَانُ انْقَطَعَ عَمَلُهُ إِلَّا مِنْ ثَلَاثَةٍ مِنْ صَدَقَةٍ جَارِيَةٍ وَعِلْمٍ يُنْتَفَعُ بِهِ وَوَلَدٍ صَالِحٍ يَدْعُو لَهُ

“Jika seseorang meninggal dunia, maka terputuslah amalannya kecuali tiga perkara (yaitu): sedekah jariyah, ilmu yang dimanfaatkan, atau do’a anak yang sholeh” (HR. Muslim no. 1631)

Hadist tersebut menyatakan bahwa ilmu yang diamalkan maka pahala tidak akan terputus. Hal tersebut juga berlaku pada ilmu yang diterapkan dalam bentuk teknologi. Seperti membuat sistem atau aplikasi yang terdapat unsur kebaikan dan kebaikan tersebut dicantumkan ke dalam aplikasi dan diamalkan oleh orang lain, maka si pembuat akan mendapat pahala seperti orang yang mengamalkan kebaikan tersebut (Hoir, 2020). Seperti halnya penelitian ini yang memberikan manfaat untuk membantu petani pada tahap pasca panen dan membuat produk yang dijual memiliki kualitas sesuai dengan kategori atau kelas yang ditawarkan dengan menggunakan sistem klasifikasi yang sudah dibuat.

Penelitian yang dilakukan Suprpto & Yulianto (2023) terkait pandangan Islam tentang pengembangan dan pemanfaatan Sainstek dalam sudut pandang bahwa teknologi digunakan sebagai media atau alat untuk mewujudkan kebutuhan hidup manusia mencakup kebutuhan primer, sekunder, ataupun tersier maka bisa dihukumi yang awalnya fardhu kifayah menjadi fardu ain, sunah, dan sebagai keutamaan. Dijelaskan pada penelitian tersebut bahwa pengembangan dan pemanfaatan teknologi dihukumi fardu ain jika teknologi tersebut sebagai syarat mutlak teralisasinya lima kebutuhan dasar primer manusia yaitu kebutuhan

terjaganya agama, jiwa, akal, keturunan, dan harta. Dan dianggap sunah bila pengembangan dan pemanfaatan teknologi dibuat sebagai penyempurna teknologi yang sudah ada.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil pengujian yang dilakukan terkait sistem klasifikasi jeruk berdasarkan tingkat kematangan dan keseragaman bentuk menggunakan data gambar dan data numerik dengan metode *Neural Network* dapat disimpulkan bahwa sistem mampu mengidentifikasi ciri unik dari setiap data dan mengklasifikasikannya namun belum terlalu sempurna yang hanya menghasilkan akurasi terbaik sebesar 77.5 %. Dan dari pengujian menggunakan 3 sistem yang berbeda dan setiap sistem diuji menggunakan kombinasi training testing yang berbeda dapat disimpulkan dalam beberapa point bahwa,

- A. Pada sistem 1 - 3 diperoleh hasil kombinasi terbaik tiap sistem dimana sistem 1 dengan perbandingan data 80:20 menghasilkan accuracy 25.0 %, precision 26.9 %, *recall* 26.9 %, dan *f-measure* 26.9 %. Sistem 2 dengan perbandingan data 70:30 menghasilkan accuracy 61.62 %, precision 60.9 %, *recall* 60.9 %, dan *f-measure* 60.9 %. Terakhir sistem 3 dengan perbandingan data 80:20 memiliki accuracy 77.5 %, precision 80.8 %, *recall* 80.8 %, dan *f-measure* 80.8 %.
- B. Berdasarkan pengamatan hasil uji coba didapatkan nilai akurasi terbaik sebesar 77.5% pada sistem 3 (80:20). Berdasarkan hasil uji coba dengan 3 sistem dan 3 uji coba kombinasi data disetiap sistem terdapat beberapa faktor yang menyebabkan variasi hasil yang berpengaruh pada kinerja sistem. Faktor pertama dalam proses pengklasifikasian kematangan menggunakan data gambar *RGB* diperlukan kompleksitas yang tinggi yang dibuktikan bahwa

sistem 3 dengan total 17 *Hidden Layer* menghasilkan akurasi tertinggi. Faktor kedua adalah persebaran jumlah *neuron* pada setiap *Hidden Layer* akhir sebelum memasuki concatenate *Layer* akan mempengaruhi hasil dalam klasifikasi kombinasi kelas seperti yang dilakukan dalam penelitian ini yang menggabungkan kelas kematangan dan kelas ukuran. Hal ini dibuktikan pada sistem 2 yang mana memiliki *Hidden Layer neuron* akhir di tiap parameter data (97,97,97,24) kurang mampu menghasilkan klasifikasi kombinasi yang baik. Dibandingkan dengan sistem 3 (97,97,97,96) mampu menghasilkan klasifikasi kombinasi lebih baik dengan sedikit kesalahan dan kesalahan rata-rata terdapat pada gagalnya mengklasifikasikan ciri kematangan data gambar. Faktor ketiga adalah jumlah data, hal ini dibuktikan pada sistem 1 dan 3 memiliki hasil terbaik pada perbandingan 80:20 karena jumlah data training lebih banyak dan pada sistem 2 memiliki selisih hasil akurasi yang tidak terlalu jauh.

5.2 Saran

Setelah uji coba yang telah dilakukan terhadap pengklasifikasian kualitas buah jeruk berdasarkan tingkat kematangan dan keseragaman bentuk menggunakan metode *Neural Network* masih terdapat ketidak sempurnaan sistem yang dibangun. Untuk peningkatan performa terkait sistem dibutuhkan pengembangan lanjutan. Berikut ini terkait saran penulis untuk pengembangan sistem selanjutnya.

A. Diperlukan peningkatan kompleksitas arsitektur dan densitas *neuron* pada tiap *Layer* terutama *Layer* yang mengolah data gambar dengan metode yang

berbeda agar dapat menangkap atau memahami kompleksitas data dengan baik serta melakukan percobaan pada hyperparameter yang digunakan.

- B. Pada tahap *preprocessing* terutama data gambar memerlukan ekstraksi fitur tambahan untuk memudahkan model mengambil informasi yang relevan dari raw data sehingga lebih mudah dalam mengidentifikasi atau memahami suatu pola dalam data yang diolah.
- C. Memperbanyak data yang digunakan agar model mampu mempelajari pola unik setiap kelas pada data yang heterogen karena jumlah kelas klasifikasi yang digunakan cukup banyak dengan 20 kelas.

DAFTAR PUSTAKA

- Abbas, A., & Suhaeti, R. N. (2016). Pemanfaatan Teknologi Pascapanen untuk Pengembangan Agroindustri Perdesaan di Indonesia. *Forum Penelitian Agro Ekonomi*, 34(1), 21. <https://doi.org/10.21082/fae.v34n1.2016.21-34>
- Abdul Aziz, M. F., Bukhari, W. M., Sukhaimie, M. N., Izzuddin, T. A., Norasikin, M. A., Rasid, A. F. A., & Bazilah, N. F. (2021). Development of Smart Sorting Machine Using Artificial Intelligence for Chili Fertigation Industries. *Journal of Automation, Mobile Robotics and Intelligent Systems*, 2021(4), 44–52. <https://doi.org/10.14313/JAMRIS/4-2021/26>
- Adhi, A. B., Anugraha, R. A., Sukma, D., & Atmaja, E. (2016). INSPEKSI VISUAL PADA SAMAK KULIT DENGAN MENERAPKAN ARTIFICIAL NEURAL NETWORK DI PT . KARYAMITRA BUDISENTOSA IMPLEMENTING ARTIFICIAL NEURAL NETWORK ON LEATHER VISUAL INSPECTION AT PT . KARYAMITRA BUDISENTOSA Grafik Waktu Proses Inspeksi Manual. *Inspeksi Visual Pada Samak Kulit Dengan Menerapkan Artificial Neural Network Di Pt. Karyamitra Budisentosa*, 3(2), 2549–2556.
- Amrizal, V., & Aini, Q. (2013). Naskah Kecerdasan Buatan. In *Kecerdasan Buatan*.
- Dalimunthe, S. R. (2019). *Analisis Perkembangan Produksi Sektor Pertanian Di Kabupaten Langkat*. <http://repository.umsu.ac.id/handle/123456789/1601>
- Denata, I., Rismawan, T., Ruslianto, I., Rekayasa Sistem Komputer, J., & MIPA Universitas Tanjungpura, F. (2021). Implementation of Deep Learning for Classification Type of Orange Using The Method Convolutional Neural Network Implementasi Deep Learning untuk Klasifikasi Jenis Jeruk dengan Menggunakan Metode Convolutional Neural Network. *Jurnal Informatika Dan Teknologi Informasi*, 18(3), 297–307. <https://doi.org/10.31515/telematika.v18i3.5541>
- Dwi, A., & Wicaksono, P. (2022). *Klasifikasi Tingkat Kematangan , Kualitas dan Jenis Buah Pisang Berdasarkan Ciri Warna dan Bentuk Menggunakan Artificial Neural Networks*. 07(02), 91–98.
- Fharadila, B., Candra, F., Bina, K., Soebrantas, H. R., Panam, S. B., Bayufharadilastudentunriacid, E., & Pendahuluan, I. (2019). *Identifikasi Kematangan Buah Nanas Menggunakan Metode Jaringan Syaraf Tiruan Mahasiswa Program Studi Teknik Informatika 2) Dosen Jurusan Teknik Informatika Program Studi Teknik Informatika S1 , Fakultas Teknik Universitas Riau Jom FTEKNIK Volume 6 Edisi . 6, 1–7*.

- Gonzalez, R. C., & Woods, R. E. (2018). *4TH EDITION Digital image processing*.
- Grandini, M., Bagli, E., & Visani, G. (2020). *Metrics for Multi-Class Classification: an Overview*. 1–17. <http://arxiv.org/abs/2008.05756>
- Gunawan, I. K. D. (2013). Klasifikasi Citra Buah Jeruk Kintamani Berdasarkan Fitur Warna dan Ukuran Menggunakan Pendekatan Euclidian Distance. *Kumpulan Artikel Mahasiswa Pendidikan Teknik Informatika*, 2(1), 261–274.
- Hanif, Z. (2020). Pengembangan Agribisnis Jeruk Nusantara. *Iptek Hortikultura*, April, 27–30.
- Hartanto, S. (2017). Implementasi Fuzzy Rule Based System. *Techsi*, 9(2), 103–117. <https://ojs.unimal.ac.id/index.php/techsi/article/view/217>
- Hoir, B. N. (2020). Sains dan Teknologi Perspektif Hadis. *Jurnal OSF Preprints*, 1–10.
- Ishaq, A. bin M. bin A. bin. (2013). *Tafsir Ibnu Katsir 8.1.pdf* (p. 570).
- Jacobs, H. R. (1987). *Geometry by Harlod R. Jacobs* (Harold R. Jacobs (ed.); Issue 48). W.H. Freeman, 1987.
- Mazen, F. M. A., & Nashat, A. A. (2019). Ripeness Classification of Bananas Using an *Artificial Neural Network*. *Arabian Journal for Science and Engineering*, 44(8), 6901–6910. <https://doi.org/10.1007/s13369-018-03695-5>
- Mohapatra, B. N. (2019). *Image edge detection techniques*. *ACCENTS Transactions on Image Processing and Computer Vision*, 5(15), 15–19. <https://doi.org/10.19101/tipcv.2019.515008>
- Patro, S. G. K., & sahu, K. K. (2015). Normalization: A *Preprocessing Stage*. *Iarjset*, 20–22. <https://doi.org/10.17148/iarjset.2015.2305>
- Purba, E. C., & Purwoko, B. S. (2019). *PENANGANAN PASCAPANEN JERUK SIAM (Citrus nobilis var. microcarpa) TUJUAN PASAR SWALAYAN*. 203–213.
- Purnama, J. A. (2020). *PERBANDINGAN ALGORITMA BACKPROPAGATION DAN SUPPORT VECTOR MACHINE PADA KLASIFIKASI TINGKAT KEMATANGAN BUAH ALPUKAT MENTEGA SKRIPSI Oleh : Jimmy Aquari Purnama Program Studi Teknik Informatika STMIK Global Informatika MDP Palembang*.
- S. Parsania, P. P., & V.Virparia, D. P. (2015). A Review: *Image Interpolation Techniques for Image Scaling*. *International Journal of Innovative Research in Computer and Communication Engineering*, 02(12), 7409–7414.

<https://doi.org/10.15680/ijirccce.2014.0212024>

- Santoso, J. T. (2021). Kecerdasan Buatan & Jaringan Syaraf Buatan. In *Penerbit Yayasan Prima Agus Teknik* (Vol. 7, Issues 1 SE-Judul Buku). <https://penerbit.stekom.ac.id/index.php/yayasanpat/article/view/177>
- Setiawan Wibisono, I., Ika Novichasari, S., & Ariesta Putri, M. (2019). “Segmentasi Fuzzy C-Means Dan *Neural Network* Untuk Membantu Identifikasi Kualitas Buah Jeruk Berdasarkan Warna Dan Ukuran.” *Jurnal Prodi Teknik Informatika UNW "Multimatrix, II(1)*, 1–5.
- Shibata, K., & Ikeda, Y. (2009). Effect of number of *Hidden neurons on learning* in large-scale *layered Neural Networks*. *ICCAS-SICE 2009 - ICROS-SICE International Joint Conference 2009, Proceedings*, 5008–5013.
- Siskandar, R., Indrawan, N. A., Kusumah, B. R., Santosa, S. H., Irmansyah, I., & Irzaman, I. (2020). Penerapan Rekayasa Mesin Sortir Sebagai Penentu Kematangan Buah Jeruk Dan Tomat Merah Berbasis *Image Processing*. *Jurnal Teknik Pertanian Lampung (Journal of Agricultural Engineering)*, 9(3), 222. <https://doi.org/10.23960/jtep-l.v9i3.222-236>
- Suprpto, A., & Yulianto, Y. (2023). Pandangan Islam Terhadap Pengembangan Dan Pemanfaatan Sains Dan Teknologi. *Es-Syajar: Journal of Islam, Science and Technology Integration*, 1(1), 1–26. <https://doi.org/10.18860/es.v1i1.20423>
- Syaputra, I. (2022). *DENGAN MENERAPKAN METODE FUZZY K-NEAREST NEIGHBOR* Ilham Syaputra , 2 Riri Syafitri Lubis , 3 Hendra Cipta Program Studi Matematika , Fakultas Sains dan Teknologi , Universitas Islam Negeri Sumatera Utara Medan. 4(1), 368–371.
- Utama, I. M. S. (2015). Prinsip Dasar Penanganan Pascapanen Buah dan Sayuran Segar. *Prinsip Dasar Penanganan Pascapanen Buah Dan Sayuran Segar.*, 1–9. https://simdos.unud.ac.id/uploads/file_pondidikan_dir/5f781f687c140f7241ed2bdf2f080013.pdf
- Wibawa, A. P., Purnama, M. G. A., Akbar, M. F., & Dwiyanto, F. A. (2018). Metode-metode Klasifikasi. *Prosiding Seminar Ilmu Komputer Dan Teknologi Informasi*, 3(1), 134.
- Yanto, B., Fimawahib, L., Supriyanto, A., Hayadi, B. H., & Pratama, R. R. (2021). Klasifikasi Tekstur Kematangan Buah Jeruk Manis Berdasarkan Tingkat Kecerahan Warna dengan Metode *Deep Learning Convolutional Neural Network*. *INOVTEK Polbeng - Seri Informatika*, 6(2), 259. <https://doi.org/10.35314/isi.v6i2.2104>

- Yanto, B., Rouza, E., Fimawahib, L., Hayadi, B. H., & Pratama, R. R. (2023). *Penerapan Algoritma Deep Learning Convolutional Neural Network Dalam Menentukan Kematangan Buah Jeruk Manis Application of the Deep Learning Convolutional Neural Network Algorithm in Determining the Murability of Sweet Orange Fruit Based on Images Red Gre.* 10(1), 59–66. <https://doi.org/10.25126/jtiik.2023105695>
- Yousuf, H., Lahzi, M., Salloum, S. A., & Shaalan, K. (2021). A systematic review on sequence-to-sequence *learning* with *Neural Network* and its models. *International Journal of Electrical and Computer Engineering*, 11(3), 2315–2326. <https://doi.org/10.11591/ijece.v11i3.pp2315-2326>