

Klasifikasi Multi-Label Jenis dan Warna Buah Menggunakan *Convolutional Neural Network (CNN)* dengan Encoder Fitur *Multi-Label Classification of Fruit Types and Colors Using Convolutional Neural Network (CNN) with Feature Encoder*

Nida Ulhasanah^{1*}, Yulison Herry Chrisnanto², Melina³, Julian Evan Chrisnanto⁴

^{1,2,3}Jurusan Informatika, Fakultas Sains dan Informatika, Universitas Jenderal Achmad Yani

⁴Fisika, Universitas Padjadjaran

¹nidaulhasanah21@if.unjani.ac.id, ²yhc@if.unjani.ac.id, ³melina@lecture.unjani.ac.id, ⁴julian20001@mail.unpad.ac.id

Abstract

Indonesia is a tropical country with a high diversity of fruits, both in terms of type and color. The main challenge in automatic fruit classification lies in the simultaneous recognition of multiple attributes under varying real-world conditions such as lighting, background, and image angles. This study aims to develop a multi-label classification model based on a Convolutional Neural Network (CNN) integrated with a ResNet-50 encoder to simultaneously recognize fruit types and colors with high accuracy and generalization. The model was trained using the diverse Fruit-360 dataset (94,110 images) and validated using 5-fold cross-validation along with data augmentation techniques to reduce the risk of overfitting. Experimental results showed that the augmented model achieved a validation accuracy of up to 97%, a precision of 98.20%, and a recall of 97.61%. The novelty of this research lies in the integration of a multi-label approach, ResNet-50 encoder, and comprehensive cross-validation within a single classification framework. Practically, this model has potential applications in image-based fruit identification systems for precision agriculture and mobile applications.

Keywords: multi-label classification, CNN, ResNet-50, Fruit-360, cross-validation, fruit classification.

Abstrak

Indonesia merupakan negara tropis dengan keanekaragaman buah yang tinggi, baik dari segi jenis maupun warna. Tantangan utama dalam klasifikasi otomatis buah adalah pengenalan atribut ganda secara simultan di bawah kondisi nyata yang bervariasi seperti pencahayaan, latar belakang, dan sudut gambar. Penelitian ini bertujuan mengembangkan model klasifikasi multi-label berbasis Convolutional Neural Network (CNN) yang dilengkapi encoder ResNet-50 untuk mengenali jenis dan warna buah secara bersamaan dengan akurasi dan generalisasi tinggi. Model dilatih menggunakan dataset Fruit-360 yang beragam (94.110 gambar), serta divalidasi menggunakan teknik 5-fold cross-validation dan augmentasi data untuk mengurangi risiko overfitting. Hasil eksperimen menunjukkan bahwa model dengan augmentasi mencapai akurasi validasi hingga 97%, precision 98,20%, dan recall 97,61%. Kebaruan dari penelitian ini adalah integrasi pendekatan multi-label, encoder ResNet-50, dan validasi silang menyeluruh dalam satu kerangka klasifikasi. Secara praktis, model ini berpotensi diterapkan dalam sistem identifikasi buah berbasis citra untuk keperluan pertanian presisi dan aplikasi mobile.

Kata kunci: klasifikasi multi-label, CNN, ResNet-50, Fruit-360, cross-validation, klasifikasi buah.

1. Pendahuluan

Pengenalan jenis Buah merupakan faktor yang cukup penting bagi masyarakat yang baru mengenal beberapa jenis buah yang jarang ditemui, sehingga seringkali menimbulkan kebingungan dalam mengidentifikasinya [1]. Indonesia memiliki keanekaragaman buah tropis yang sangat melimpah, baik dari segi jenis maupun warna. Terdapat sekitar 295.383 spesies tumbuhan berbiji yang dapat menghasilkan buah [2]. Namun, potensi besar ini belum dikelola secara optimal karena masih terbatasnya sistem identifikasi dan klasifikasi buah yang dapat diandalkan dalam skala besar [3].

Dalam beberapa tahun terakhir, perkembangan teknologi kecerdasan buatan (AI) telah mendorong penerapan metode klasifikasi otomatis dalam pengolahan citra digital di berbagai bidang, termasuk pertanian dan pangan [4]. Salah satu cabang dari AI yang memainkan peran penting adalah deep learning, terutama melalui arsitektur pembelajaran bertingkat seperti Convolutional Neural Network (CNN), yang telah banyak dimanfaatkan dalam berbagai penelitian. CNN terbukti efektif dalam mengekstraksi pola visual dari citra dan menjadi fondasi berbagai sistem klasifikasi citra terkini. Selain itu, penggunaan teknik transfer learning dengan arsitektur seperti ResNet-50

telah meningkatkan presisi klasifikasi citra dalam banyak studi [5].

Beberapa penelitian sebelumnya menunjukkan efektivitas pendekatan ini. Penelitian yang dilakukan oleh [6] mengusulkan model CNN untuk klasifikasi empat jenis buah jeruk menggunakan 40 gambar dan berhasil mencapai akurasi sebesar 95%. Namun, keterbatasan utama dari studi ini terletak pada ukuran dataset yang sangat kecil dan kurangnya variasi kondisi lingkungan gambar, seperti pencahayaan dan latar belakang. Selain itu, pendekatan lanjutan seperti transfer learning atau fine-tuning belum diterapkan.

Penelitian lain yang dilakukan oleh [7] menunjukkan potensi CNN dalam mengklasifikasikan tingkat kesegaran buah ke dalam enam kelas menggunakan dataset dari Kaggle, dengan akurasi rata-rata tertinggi sebesar 93%. Meskipun cukup menjanjikan, model ini mengalami 91 kesalahan klasifikasi yang mengindikasikan perlunya peningkatan pada struktur model dan representasi fitur yang lebih baik

Sementara itu, penelitian lain oleh [8] menggunakan arsitektur *ResNet-50* dengan pendekatan *transfer learning* untuk klasifikasi citra bibit kelapa sawit ke dalam empat kelas, termasuk penyakit dan hama. Dataset awal sebanyak 612 gambar diperbesar menjadi 1.749 gambar melalui augmentasi untuk mencegah overfitting. Hasil pengujian menunjukkan akurasi hingga 95%, dengan evaluasi menggunakan akurasi, precision, recall, dan confusion matrix. Hal ini membuktikan efektivitas *ResNet-50* dalam mendeteksi fitur visual pada citra tumbuhan secara mendalam.

Dari penelitian - penelitian sebelumnya, dapat disimpulkan bahwa CNN memiliki potensi besar dalam klasifikasi buah dan tanaman berdasarkan jenis, kesegaran, dan kondisi kesehatan. Namun, kebanyakan masih terbatas pada klasifikasi label tunggal, skala dataset kecil, dan belum menguji generalisasi model secara menyeluruh. Padahal, dalam kondisi nyata, objek sering kali memiliki lebih dari satu atribut penting seperti jenis dan warna. Pendekatan multi-label konvensional sering kali mengabaikan keterkaitan antar label yang berdampak pada akurasi prediksi. Menjawab celah tersebut, dibutuhkan pendekatan yang lebih komprehensif melalui klasifikasi multi-label, pemanfaatan arsitektur yang lebih kuat seperti *ResNet-50*, dan validasi menyeluruh menggunakan *n-fold cross-validation* [9].

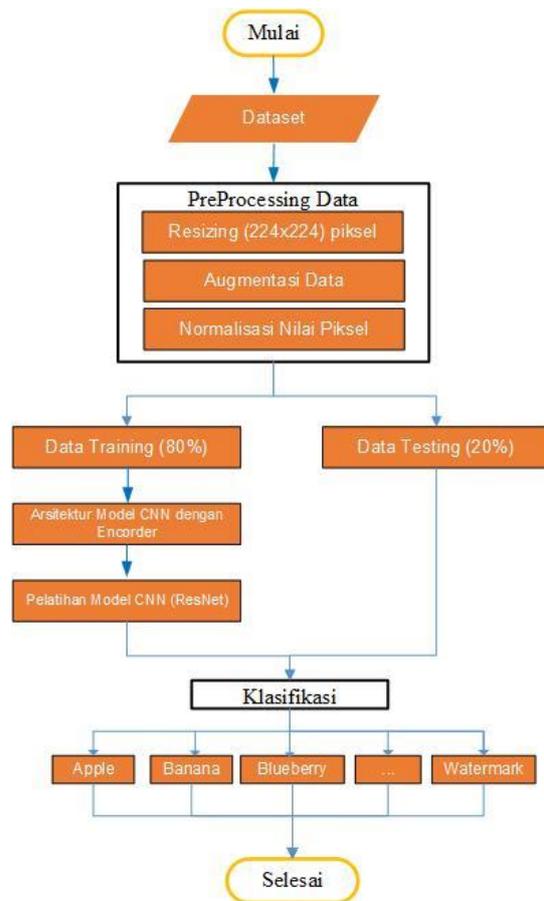
Keterbatasan tersebut menjadi dasar dilakukannya penelitian ini, yaitu untuk mengembangkan model klasifikasi multi-label buah yang mampu mengenali jenis dan warna secara bersamaan dengan akurasi tinggi dan generalisasi yang baik. Dataset Fruit-360 yang berskala besar dan memiliki variasi tinggi digunakan bersama dengan arsitektur CNN yang memanfaatkan encoder *ResNet-50* yang dikenal kuat dalam ekstraksi

fitur visual. Untuk mengurangi risiko overfitting dan mengevaluasi performa model secara menyeluruh, diterapkan teknik *n-fold cross-validation* [10].

Penelitian ini mengkaji pengaruh penggunaan dataset yang besar dan bervariasi serta penerapan CNN dengan encoder *ResNet-50* terhadap kinerja model klasifikasi multi-label buah. Selain itu, penelitian ini mengevaluasi kontribusi penerapan *n-fold cross-validation* dalam meningkatkan validitas dan generalisasi model. Keunikan pendekatan ini terletak pada integrasi multi-label classification, *ResNet-50*, dan validasi silang menyeluruh dalam satu kerangka yang belum banyak diulas dalam penelitian terdahulu.

2. Metode Penelitian

Penelitian ini merupakan studi eksperimental dalam bidang pengolahan citra digital yang bertujuan untuk mengembangkan pendekatan klasifikasi multi-label berbasis *deep learning* menggunakan arsitektur CNN. Pendekatan ini dirancang untuk mampu mengidentifikasi dua label sekaligus dari gambar buah, yaitu jenis dan warna, dengan mengintegrasikan encoder sebagai bagian dari proses ekstraksi fitur guna meningkatkan akurasi dan efisiensi klasifikasi. Proses penelitian dilakukan secara sistematis melalui beberapa tahapan yang dijelaskan pada Gambar 1.



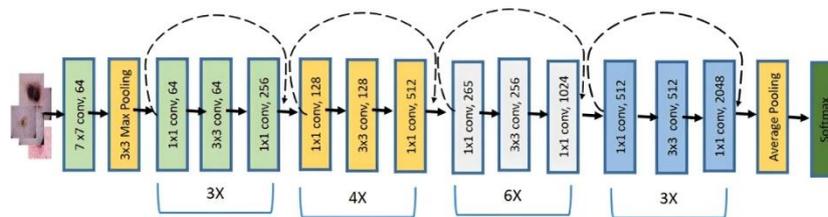
Gambar 1. Alur Penelitian

Penelitian ini dimulai dengan pengumpulan dataset Fruit-360 yang terdiri dari berbagai gambar buah-buahan. Dataset ini diperoleh dari platform Kaggle versi 2024.08.04.0 dan mencakup 94.110 gambar dengan resolusi awal 100×100 piksel. Dari dataset ini, dipilih 15 kelas buah yang digunakan dalam proses pelatihan dan pengujian model yaitu *Apple, Banana, Blueberry, Cherry, Grape, Granadilla, Kiwi, Lemon, Mango, Orange, Papaya, Pear, Pomegranate, Strawberry, dan Watermelon*.

Pada tahap kedua, setelah pengumpulan data, dilakukan tahapan praproses yang meliputi perubahan ukuran citra menjadi dimensi seragam, normalisasi nilai piksel, serta augmentasi data menggunakan rotasi, dan *flipping*

[2]. Langkah ini bertujuan untuk memperkaya variasi data pelatihan dan mengurangi risiko *overfitting* pada model. Terakhir, dataset dibagi menjadi dua subset utama, yaitu data latih (80%) dan data uji (20%). Data latih digunakan untuk mengajarkan model mengenali pola dalam gambar, sedangkan data validasi digunakan untuk memantau kinerja model selama pelatihan serta melakukan tuning hyperparameter [6], [11].

Tahap ketiga adalah Perancangan Arsitektur model. Arsitektur yang digunakan dalam penelitian ini adalah CNN dengan basis encoder *ResNet-50*, yang diadopsi melalui pendekatan *transfer learning*. Arsitektur *ResNet-50* dapat dilihat pada Gambar 2.

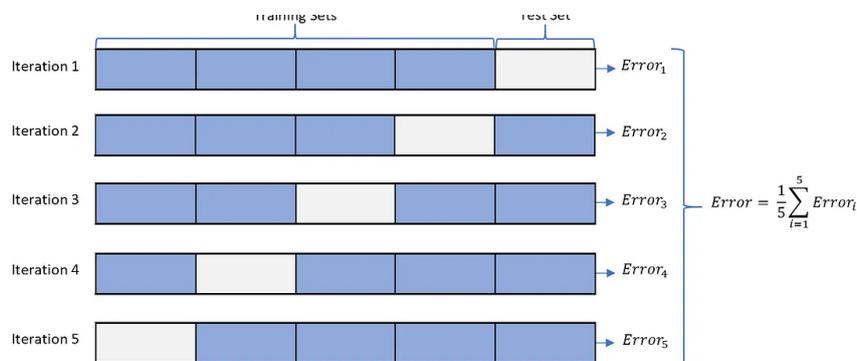


Gambar 2. Arsitektur *ResNet-50*

ResNet-50 dipilih karena kemampuannya dalam mengekstraksi fitur visual secara mendalam melalui 50 lapisan jaringan, serta dukungan struktur residual blocks yang dapat mengatasi permasalahan vanishing gradient dan memungkinkan propagasi informasi lebih efisien pada jaringan yang dalam [2], [8].

Pada tahap keempat, untuk meningkatkan reliabilitas hasil dilakukan *n-fold cross-validation* sebagai teknik

validasi model. Metode ini membagi dataset menjadi k bagian (fold), dimana setiap fold digunakan sebagai data validasi satu kali, sementara sisanya digunakan sebagai data latih. Proses ini diulang sebanyak n kali, sehingga setiap bagian dataset digunakan sebagai data validasi pada satu iterasi. Hasil dari *n-fold cross-validation*, kemudian dirata-ratakan untuk memberikan gambaran yang lebih akurat tentang performa model [12].

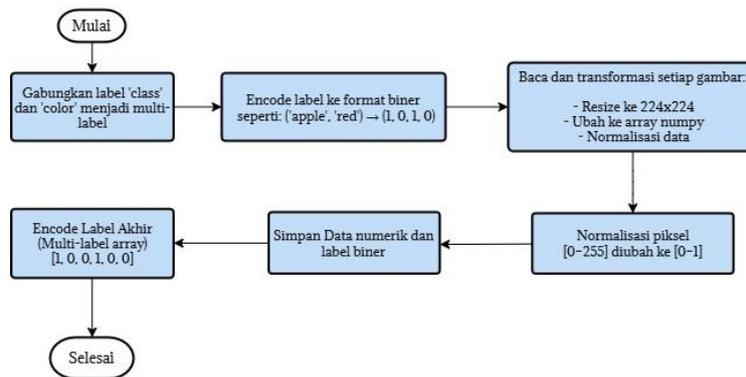


Gambar 3. Contoh *Cross Validation*

Pelatihan model dalam penelitian ini dilakukan menggunakan framework TensorFlow 2.19.0 dan Keras API, yang diimplementasikan dalam lingkungan pemrograman Python 3.12.6 pada sistem operasi Windows 64-bit. Seluruh proses pengembangan, pelatihan, dan evaluasi model dijalankan pada perangkat dengan spesifikasi CPU Intel Core i3, RAM 32 GB, serta GPU onboard Intel UHD Graphics. Meskipun tanpa dukungan GPU terdedikasi, proses pelatihan tetap dapat berjalan secara efisien dengan

memanfaatkan teknik pelatihan bertahap dan batch yang disesuaikan dengan kapasitas perangkat keras.

Sebelum digunakan dalam pelatihan model multi-label, data gambar dan label perlu ditransformasikan ke format numerik. Proses ini mencakup penggabungan label kelas dan warna, encoding ke format biner, transformasi gambar menjadi array numerik (resize, konversi NumPy, dan normalisasi piksel), serta penyimpanan hasil akhir. Alur proses tersebut dapat dilihat pada Gambar 4.



Gambar 4. Flowchart Proses Transformasi Data

Setelah menyelesaikan perancangan model, langkah berikutnya adalah tahap kelima yaitu melakukan pelatihan model. Dilakukan dengan menggunakan dua skenario berbeda: satu dengan augmentasi data dan satu lagi tanpa augmentasi data. Setiap skenario dilatih menggunakan *batch size* sebesar 32 dan jumlah *epoch* yang ditentukan berdasarkan konvergensi model pada dataset validasi. Proses pelatihan dimulai dengan mengoptimalkan parameter model pada data latih, sedangkan kinerja model selama pelatihan dimonitor menggunakan data validasi. Untuk mencegah *overfitting*, teknik *EarlyStopping* diterapkan untuk menghentikan pelatihan lebih awal jika tidak ada perbaikan dalam validation loss setelah beberapa *epoch* berturut-turut [13].

Pada tahap keenam yaitu klasifikasi, model dikonfigurasi untuk mengenali 15 kelas buah yang dipilih dari dataset. Model ini menggunakan fungsi aktivasi *Sigmoid* pada lapisan output karena tugas yang dihadapi adalah klasifikasi multi-kelas, di mana setiap gambar hanya dapat memiliki satu label dari beberapa kelas, seperti jenis buah dan warnanya. Untuk mengoptimalkan proses pelatihan, digunakan optimizer *Adam*, yang memiliki keunggulan dalam menyesuaikan laju pembelajaran dan mempercepat konvergensi model [14].

Pada tahap ketujuh, model yang sudah dilatih, kinerjanya dievaluasi menggunakan *precision* untuk menganalisis jumlah prediksi benar dan salah pada setiap kelas. Hal ini bertujuan untuk memahami pola kesalahan klasifikasi yang terjadi pada model. Sebagai bagian dari evaluasi, metrik seperti akurasi, *precision*, *recall*, dan *F1-score* digunakan untuk mengukur performa model dalam mengenali kelas buah yang dipilih. Perbandingan dilakukan antara model yang dilatih dengan teknik augmentasi dan model tanpa augmentasi untuk menganalisis pengaruh augmentasi data terhadap kinerja model klasifikasi [14], [15].

Untuk model dengan augmentasi, *confusion matrix* dihitung berdasarkan prediksi yang dihasilkan oleh model pada data validasi yang berasal dari *val_generator*. *Confusion matrix* adalah salah satu

metode yang dapat digunakan untuk mengevaluasi kinerja suatu model klasifikasi dengan menganalisis nilai akurasi dari model tersebut [16]. *Confusion matrix* dapat dilihat pada Gambar 5.

		Predicted Class	
		Positive	Negative
Actual Class	Positive	TP	FN
	Negative	FP	TN

Gambar 5. *Confusion matrix*

Berdasarkan *confusion matrix*, dapat dilakukan perhitungan nilai akurasi, yaitu true positif (TP), true negatif (TN), false positif (FP), dan false negatif (FN). Keempat parameter ini menjadi dasar penentuan metrik performa seperti akurasi, presisi, *recall*, dan *F1-score*.

Akurasi adalah proporsi prediksi yang benar terhadap seluruh data, yang dihitung dari jumlah True Positive (TP) dan True Negative (TN) dibagi dengan jumlah total data. Nilai akurasi memberikan gambaran umum tentang seberapa baik model dalam melakukan klasifikasi secara keseluruhan, baik untuk kelas positif maupun negative [17]. Rumus untuk menghitung akurasi ditunjukkan pada Persamaan 1.

$$\text{Akurasi} = \frac{TP+TN}{TP+FP+FN+TN} \quad (1)$$

Presisi merupakan metrik evaluasi yang digunakan untuk mengukur seberapa banyak prediksi positif yang benar (True Positive/TP) dibandingkan dengan seluruh prediksi yang diklasifikasikan sebagai positif, termasuk prediksi yang salah (False Positive/FP) [18]. Rumus presisi yang digunakan dalam evaluasi model disajikan pada Persamaan 2.

$$\text{Precision} = \frac{TP}{TP + FP} \quad (2)$$

Recall adalah rasio true positif dibandingkan dengan semua data positif. Nilai *recall* menunjukkan seberapa baik model dalam mendeteksi semua data yang benar-benar termasuk dalam kelas positif, atau seberapa

sensitif model terhadap data dari kelas tersebut [19]. Perhitungan *recall* dapat dilihat pada Persamaan 3.

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

Sedangkan *F-1 score* adalah merepresentasikan pembobotan perbandingan nilai rata-rata antara nilai *recall* dan nilai *precision* [20]. Perhitungan *F-1 score* dapat dilihat pada Persamaan 4.

$$f1 - score = \frac{=2*Precision*Recall}{Precision+Recall} \quad (4)$$

Selanjutnya, hasil *confusion matrix* divisualisasikan menggunakan heatmap dan dianalisis lebih lanjut melalui *classification_report* untuk memperoleh metrik evaluasi yang lebih mendalam. Selain itu, dilakukan visualisasi perbandingan akurasi dan loss sepanjang *epoch* antara kedua model. Grafik perbandingan ini memberikan gambaran tentang pengaruh augmentasi data terhadap kinerja model, khususnya dalam hal peningkatan akurasi dan pengurangan loss selama pelatihan.

3. Hasil dan Pembahasan

3.1 Hasil Preprocessing Data

Proses preprocessing data bertujuan untuk mempersiapkan dataset yang akan digunakan dalam model klasifikasi multi-label untuk prediksi jenis buah dan warna. Terdapat 5 tahapan yang dilakukan dalam proses preprocessing.

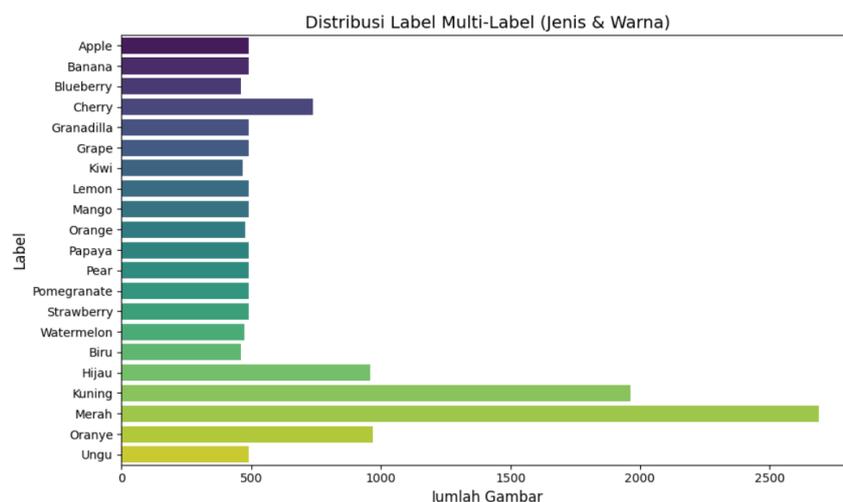
Pemilihan dan Penyaringan Data: Pada tahap ini, dataset yang digunakan adalah Fruit-360, yang berisi gambar dari berbagai jenis buah. Berdasarkan tujuan penelitian, data yang digunakan dibatasi hanya pada 15

jenis buah, yaitu: *Apple, Banana, Blueberry, Cherry, Grape, Granadilla, Kiwi, Lemon, Mango, Orange, Papaya, Pear, Pomegranate, Strawberry, dan Watermelon*. Setiap buah diberikan label warna sesuai kategori yang telah ditentukan, seperti Merah untuk *Apple* dan *Strawberry*, Kuning untuk *Banana* dan *Mango*, serta Hijau untuk *Kiwi* dan *Pear*. Data gambar kemudian disaring sesuai dengan kategori buah yang telah dipilih dan dimasukkan ke dalam dataframe bersama label jenis buahnya.

Penambahan Label Warna: Setelah data gambar disaring, langkah selanjutnya adalah menambahkan informasi label warna untuk setiap jenis buah. Label warna diberikan berdasarkan kamus *fruit_colors* yang telah ditentukan sebelumnya. Setiap gambar diberi label warna sesuai dengan jenis buahnya, yang kemudian dikombinasikan dengan label jenis buah dalam kolom baru yang disebut labels.

Binarisasi Label: Untuk mempersiapkan data bagi model klasifikasi multi-label, dilakukan binarisasi terhadap label jenis buah dan warna. Proses ini dilakukan menggunakan *MultiLabelBinarizer*, yang mengubah kategori buah dan warna menjadi format biner yang dapat diproses oleh model pembelajaran mesin. Hasil binarisasi menghasilkan dua set label terpisah: satu untuk jenis buah dan satu untuk warna.

Visualisasi Distribusi Label: Untuk memastikan distribusi data yang seimbang antara kelas, dilakukan visualisasi terhadap jumlah kemunculan setiap label jenis buah dan warna. Distribusi jumlah gambar di setiap kelas buah divisualisasikan menggunakan diagram batang, yang bisa dilihat pada Gambar 6.



Gambar 6. Distribusi Multi-Label (Jenis dan Warna)

Dari hasil visualisasi tersebut, terlihat bahwa distribusi gambar antar kelas cukup seimbang, meskipun ada beberapa kelas yang sedikit lebih banyak gambar daripada yang lain. Dengan distribusi ini, model

memiliki kesempatan untuk mempelajari ciri khas setiap buah dengan baik.

Selain itu, untuk memastikan bahwa dataset ini cukup bervariasi, contoh gambar dari masing-masing kelas

buah juga ditampilkan. Contoh gambar ini menggambarkan visualisasi dari setiap kategori buah seperti Apel, Pisang, Blueberry, dan lain-lain, yang disajikan dalam bentuk subplot pada Gambar 7.



Gambar 7. Kategori Buah

3.2 Hasil Perancangan Model

Pada tahap perancangan model, dilakukan pendekatan klasifikasi multi-label menggunakan arsitektur *Convolutional Neural Network* (CNN) yang dikombinasikan dengan encoder *ResNet-50* sebagai *feature extractor*. Dataset yang digunakan telah diproses terlebih dahulu dengan mengubah label menjadi format multi-label biner menggunakan *MultiLabelBinarizer*.

Data kemudian dibagi menggunakan teknik Stratified n-Fold Cross Validation sebanyak lima lipatan (5-fold) guna memastikan pemerataan distribusi label pada setiap pelatihan dan validasi. Model dibangun dengan memanfaatkan *ResNet-50* pretrained dari ImageNet, dengan lapisan atas dihilangkan (*include_top=False*) dan beberapa lapisan akhir dibuka untuk pelatihan ulang (*fine-tuning*) guna meningkatkan akurasi model terhadap data baru.

Setelah lapisan encoder, ditambahkan beberapa lapisan tambahan berupa *GlobalAveragePooling*, *BatchNormalization*, *Dropout*, dan *Dense* dengan aktivasi *ReLU*, serta output layer menggunakan aktivasi *sigmoid* agar mendukung multi-label. Model dikompilasi menggunakan optimizer *Adam* dengan *learning rate* kecil untuk mendukung *fine-tuning*, serta menggunakan *loss function binary_crossentropy* yang sesuai untuk tugas klasifikasi multi-label.

3.3 Hasil Pelatihan Model

Pada hasil pelatihan, penelitian ini bertujuan untuk mengevaluasi performa model *Convolutional Neural Network* (CNN) berbasis *ResNet-50* dalam klasifikasi gambar buah, dengan membandingkan dua skenario: model dengan augmentasi data dan model tanpa augmentasi data. Pelatihan dilakukan menggunakan dataset gambar berukuran 224×224 piksel, dengan *batch size* 32 selama 15 *epoch*. Proses pelatihan ini

menilai kinerja model berdasarkan metrik akurasi, *precision*, *recall*, dan *loss* pada data latih dan validasi. Untuk mendukung pelatihan yang optimal dan mencegah *overfitting*, diterapkan strategi *early stopping* dan model *checkpoint*.

Sebagai upaya untuk memperoleh penilaian kinerja model yang lebih menyeluruh dan bebas dari bias pembagian data, penelitian ini menggunakan pendekatan *5-Fold Cross-Validation*. Metode ini membagi dataset ke dalam lima bagian yang seimbang, kemudian secara bergiliran masing-masing bagian digunakan sebagai data validasi, sementara sisanya berfungsi sebagai data pelatihan. Proses ini diulang sebanyak lima kali, sehingga seluruh data mendapatkan kesempatan yang adil untuk diuji.

Tabel 1 menunjukkan hasil evaluasi untuk setiap fold yang digunakan dalam *cross-validation*, meliputi *F1-score*, *Precision*, *Recall*, dan *Loss* pada data validasi untuk setiap fold:

Tabel 1. Hasil n-fold cross-validation

Fold	<i>F1-score</i>	<i>Precision</i>	<i>Recall</i>	<i>Loss</i>
1	0.8271	0.8496	0.8045	0.1342
2	0.8394	0.8612	0.8168	0.1228
3	0.8512	0.8735	0.8296	0.1105
4	0.8410	0.8623	0.8201	0.1186
5	0.8436	0.8644	0.8232	0.1163

Berdasarkan pengujian tersebut, diperoleh rata-rata *F1-score* sebesar 0.8405, *precision* sebesar 0.8622, *recall* sebesar 0.8188, dan *loss* sebesar 0.1205. Nilai-nilai ini mencerminkan bahwa model tidak hanya mampu mengenali label ganda dengan baik, tetapi juga memiliki tingkat generalisasi yang tinggi serta ketahanan terhadap *overfitting* selama pelatihan.

3.4 Hasil Evaluasi Model

Model dengan augmentasi data menunjukkan kinerja yang lebih stabil sepanjang pelatihan. Akurasi pada data latih meningkat secara konsisten, yang mengindikasikan bahwa model terus belajar dengan baik dari data yang diberikan. Sementara itu, *loss* pada data validasi terus menurun, menunjukkan bahwa teknik augmentasi berhasil membantu model dalam menghindari *overfitting* dan meningkatkan kemampuannya untuk menggeneralisasi pola yang ada dalam data.

Pada akhir pelatihan, model mencapai akurasi validasi tertinggi sekitar 87%. Selain itu, model juga mencatatkan nilai *precision* sebesar 98.20% dan *recall* 97.61%, yang menunjukkan kemampuannya dalam mengenali dan mengklasifikasikan data dengan akurat. Tabel 2 adalah hasil perbandingan metrik antara data latih dan data validasi untuk model dengan augmentasi:

Tabel 2. Model dengan Augmentasi

Metrik	Data Latih	Data Validasi
Akurasi	98.39%	87.88%
Presisi	98.63%	98.20%
Recall	98.10%	97.61%

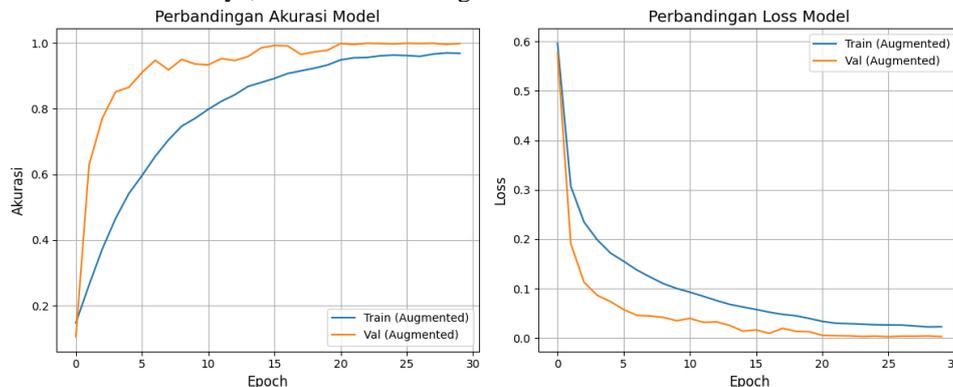
Perbandingan metrik antara model dengan dan tanpa augmentasi dapat dilihat pada Tabel 3:

Tabel 3. Model tanpa Augmentasi

Metrik	Data Latih	Data Validasi
Akurasi	54.35%	56.14%
Presisi	95.08%	95.08%
Recall	6.37%	7.24%

Sebaliknya, model tanpa augmentasi data menunjukkan kinerja yang jauh lebih rendah. Meskipun akurasi pada data latih terus meningkat, model mulai menunjukkan tanda-tanda *overfitting* setelah *epoch* ke-10, yang terlihat dari peningkatan nilai loss pada data validasi meskipun akurasi pada data latih terus meningkat. Hal ini menunjukkan bahwa model tidak dapat menggeneralisasi dengan baik pada data baru. Akurasi validasi model tanpa augmentasi hanya mencapai sekitar 56%, yang menunjukkan bahwa performa model pada data validasi jauh lebih rendah dibandingkan dengan pada data latih.

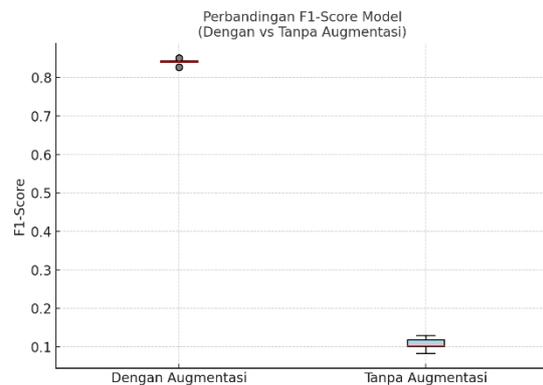
Precision pada model ini tercatat sebesar 95.08%, namun *recall* sangat rendah, hanya 7.24%, yang menandakan bahwa model kesulitan dalam mengenali sebagian besar kelas dengan akurat. Nilai recall yang sangat rendah ini dapat dijelaskan oleh adanya dominasi kelas tertentu dalam dataset, yang menyebabkan model bias terhadap kelas mayoritas. Tanpa augmentasi, distribusi visual antar kelas menjadi kurang beragam, sehingga model tidak cukup terpapar pada variasi fitur dari kelas minoritas. Akibatnya, model cenderung



Gambar 9. Grafik Perbandingan akurasi model dan loss model

Untuk mengevaluasi kemampuan model dalam klasifikasi pada masing-masing kelas, dibuatlah confusion matrix berdasarkan hasil prediksi terhadap data validasi. Confusion matrix pada Gambar 10 menunjukkan bahwa elemen diagonal yang menandakan prediksi benar dominan pada beberapa kelas seperti Apple, Banana, dan Cherry. Namun, terdapat kesalahan pada beberapa elemen non-diagonal, seperti kelas Blueberry dan Pomegranate, yang

mengabaikan kelas-kelas yang jarang muncul atau yang memiliki kemiripan visual dengan kelas mayoritas, yang secara langsung menurunkan kemampuan deteksi (*recall*) terhadap kelas-kelas tersebut.



Gambar 8. F1-Score Model dengan dan tanpa Augmentasi

Untuk menguji signifikansi perbedaan performa antara kedua model, dilakukan uji statistik *paired t-test* terhadap nilai F1-score dari masing-masing fold yang dapat dilihat pada Gambar 8. Hasil uji menunjukkan nilai *t-statistic* sebesar 96,43 dan *p-value* sebesar $6,93 \times 10^{-8}$, yang berarti bahwa perbedaan performa antara kedua model signifikan secara statistik ($p < 0.05$).

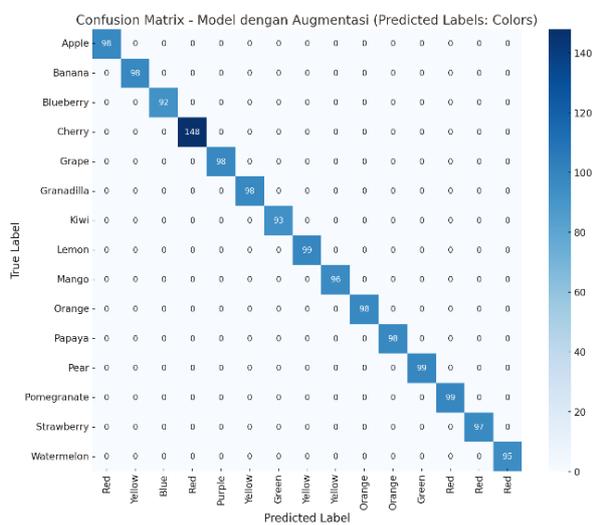
Perbandingan grafik akurasi dan loss antara kedua model dapat dilihat pada Gambar 9. Model dengan augmentasi data menunjukkan peningkatan akurasi yang stabil dan penurunan loss yang konsisten, sementara model tanpa augmentasi data mengalami fluktuasi besar pada loss dan tidak menunjukkan perbaikan yang berarti dalam akurasi.

kemungkinan disebabkan oleh kemiripan visual antar kelas, variasi pencahayaan, atau tekstur gambar.

Secara keseluruhan, model dengan augmentasi data memiliki performa yang jauh lebih baik dibandingkan model tanpa augmentasi. Model dengan augmentasi menunjukkan peningkatan yang konsisten dalam akurasi, dengan nilai validasi yang mencapai 87% dan nilai *precision* serta *recall* yang sangat tinggi. Di sisi

lain, model tanpa augmentasi mengalami *overfitting*, dengan akurasi validasi hanya mencapai 56%, *precision* yang lebih rendah, dan *recall* yang sangat rendah.

Dengan hasil performa yang tinggi pada model yang menggunakan augmentasi data, pendekatan klasifikasi multi-label ini punya peluang besar untuk diterapkan dalam kehidupan sehari-hari. Salah satu contohnya adalah lewat aplikasi mobile berbasis kamera, yang bisa membantu pengguna seperti konsumen, pedagang, atau pelaku UMKM dalam mengenali jenis dan warna buah secara otomatis. Di bidang pertanian dan logistik, model ini juga bisa dimanfaatkan untuk menyortir buah secara otomatis di gudang pascapanen atau mengecek kualitas buah lewat tampilan visual. Karena model ini mampu mengenali lebih dari satu atribut sekaligus seperti jenis dan warna. Pendekatan ini cocok digunakan di situasi nyata yang kompleks, misalnya di lini produksi atau sistem pertanian berbasis teknologi visual.



Gambar 10. Confusion matrix

4. Kesimpulan

Berdasarkan hasil penelitian ini, dapat disimpulkan bahwa augmentasi data secara signifikan meningkatkan performa model klasifikasi multi-label dalam mengidentifikasi jenis dan warna buah menggunakan arsitektur CNN berbasis ResNet-50. Model yang dilatih dengan augmentasi data mencapai akurasi validasi hingga 87.88%, dengan nilai *precision* sebesar 98.20% dan *recall* sebesar 97.61%, yang jauh lebih baik dibandingkan model tanpa augmentasi. Model tanpa augmentasi hanya mencapai akurasi validasi sebesar 56.14% berdasarkan hasil pelatihan per fold. Namun, ketika dievaluasi secara keseluruhan pada data validasi akhir, akurasinya hanya mencapai 3.58%. Hal ini menunjukkan bahwa tanpa augmentasi, model sangat rentan terhadap *overfitting* dan gagal menggeneralisasi data baru dengan baik. Penerapan teknik *cross-validation* dan augmentasi data terbukti membantu

mengatasi masalah *overfitting* serta meningkatkan generalisasi model. Kontribusi utama dari penelitian ini adalah integrasi pendekatan multi-label, arsitektur ResNet-50, serta evaluasi komprehensif berbasis *cross-validation* dan uji statistik dalam satu kerangka klasifikasi yang kuat.

Daftar Rujukan

- [1] Edwin Febrywinata, "Pengenalan Dan Klasifikasi Jenis Buah Menggunakan Metode CNN Secara Sederhana Dengan Menggunakan Google Colab," *Merkurius J. Ris. Sist. Inf. dan Tek. Inform.*, vol. 2, no. 4, pp. 185–193, 2024, doi: 10.61132/merkurius.v2i4.162.
- [2] S. Noris and A. Waluyo, "Penerapan Deep Learning untuk Klasifikasi Buah Menggunakan Algoritma Convolutional Neural Network (CNN)," *J. Teknol. Sist. Inf. dan Apl.*, vol. 6, no. 1, pp. 39–46, 2023, doi: 10.32493/jtsi.v6i1.29648.
- [3] R. Namruddin, Mirfan, and Irfandi, "Klasifikasi Kesegaran Buah Apel Menggunakan Metode Convolutional Neural Network (CNN) Berbasis Android," *Pros. SISFOTEK*, pp. 295–302, 2023.
- [4] Anissa Ollivia Cahya Pratiwi, "Klasifikasi Jenis Anggur Berdasarkan Bentuk Daun Menggunakan Convolutional Neural Network Dan K-Nearest Neighbor," *J. Ilm. Tek. Inform. dan Komun.*, vol. 3, no. 2, pp. 201–224, 2023, doi: 10.55606/juitik.v3i2.535.
- [5] Handoko Adji Pangestu and Kusri, "Peningkatan kinerja arsitektur ResNet50 untuk Menangani Masalah *Overfitting* dalam Klasifikasi Penyakit Kulit," *Tematik*, vol. 11, no. 1, pp. 65–71, 2024, doi: 10.38204/tematik.v11i1.1876.
- [6] Yazid Fauzan Nur Ashfani, Yovi Litanianda, and Rizqy Amalia Putri, "Klasifikasi Jenis Buah Jeruk Menggunakan Metode Convolutional Neural Network: Deep Learning Studi," *Uranus J. Ilm. Tek. Elektro, Sains dan Inform.*, vol. 2, no. 2, pp. 70–79, 2024, doi: 10.61132/uranus.v2i2.129.
- [7] F. Paraijun, R. N. Aziza, and D. Kuswardani, "Implementasi Algoritma Convolutional Neural Network Dalam Mengklasifikasi Kesegaran Buah Berdasarkan Citra Buah," *Kilat*, vol. 11, no. 1, pp. 1–9, 2022, doi: 10.33322/kilat.v10i2.1458.
- [8] E. Oktafanda, "Klasifikasi Citra Kualitas Bibit dalam Meningkatkan Produksi Kelapa Sawit Menggunakan Metode Convolutional Neural Network (CNN)," *J. Inform. Ekon. Bisnis*, vol. 4, no. 3, pp. 72–77, 2022, doi: 10.37034/infek.v4i3.143.
- [9] X. Liu and Y. Hu, "Multi-Label Image Classification Based on Object Detection and Dynamic," 2024, doi: 10.32604/cmc.2024.053938.
- [10] P. Sukhetha, N. Hemalatha, and R. Sukumar, "Classification of fruits and vegetables using ResNet model.," *agriRxiv*, vol. 2021, no. August, 2021, doi: 10.31220/agriRxiv.2021.00075.
- [11] E. Setia Budi, A. Nofriyaldi Chan, P. Priscillia Alda, and M. Arif Fauzi Idris, "RESOLUSI: Rekayasa Teknik Informatika dan Informasi Optimasi Model Machine Learning untuk Klasifikasi dan Prediksi Citra Menggunakan Algoritma Convolutional Neural Network," *Media Online*, vol. 4, no. 5, p. 509, 2024, [Online]. Available: <https://djournal.com/resolusi>
- [12] F. Muhammad, A. M. Arimurthy, and D. Chahyati, "Transfer Learning dengan Metode Fine Tuning pada Model Network VGG16 dan ResNet50," *Indones. J. Comput. Sci. Attrib.*, vol. 12, no. 1, pp. 361–374, 2023.
- [13] B. Dan, S. Di, and P. Tradisional, "IMPLEMENTASI CNN RESNET50 UNTUK MENDETEKSI KUALITAS," vol. 9, no. 3, pp. 3675–3682, 2025.
- [14] T. Bariyah, M. A. Rasyidi, and N. Ngatini, "Convolutional Neural Network untuk Metode Klasifikasi Multi-Label pada Motif Batik," *Techno.Com*, vol. 20, no. 1, pp. 155–165, 2021, doi: 10.33633/tc.v20i1.4224.

- [15] Y. Jumaryadi, A. Muhammad Ihsan, and B. Priambodo, "KLIK: Kajian Ilmiah Informatika dan Komputer Klasifikasi Jenis Buah-Buahan Menggunakan Citra Digital Dengan Metode Convolutional Neural Networks," *Media Online*, vol. 4, no. 3, pp. 1737–1746, 2023, doi: 10.30865/klik.v4i3.1421.
- [16] S. Dewi, F. Ramadhani, and S. Djasmayena, "Klasifikasi Jenis Jerawat Berdasarkan Gambar Menggunakan Algoritma CNN (Convolutional Neural Network)," *Hello World J. Ilmu Komput.*, vol. 3, no. 2, pp. 68–73, 2024, doi: 10.56211/helloworld.v3i2.518.
- [17] P. Dewi, P. Purwono, and S. Kurniawan Dwi, "Pemanfaatan Teknologi Machine Learning pada Klasifikasi Jenis Hipertensi Berdasarkan Fitur Pribadi," *Smart Comp Jurnalnya Orang Pint. Komput.*, vol. 11, no. 3, pp. 377–387, 2022, doi: 10.30591/smartcomp.v11i3.3721.
- [18] R. R. Waliyansyah and C. Fitriyah, "Perbandingan Akurasi Klasifikasi Citra Kayu Jati Menggunakan Metode Naive Bayes dan k-Nearest Neighbor (k-NN)," *J. Edukasi dan Penelit. Inform.*, vol. 5, no. 2, p. 157, 2019, doi: 10.26418/jp.v5i2.32473.
- [19] W. Hidayat, M. Ardiansyah, and A. Setyanto, "Pengaruh Algoritma ADASYN dan SMOTE terhadap Performa Support Vector Machine pada Ketidakseimbangan Dataset Airbnb," *Edumatic J. Pendidik. Inform.*, vol. 5, no. 1, pp. 11–20, 2021, doi: 10.29408/edumatic.v5i1.3125.
- [20] A. B. Prakosa, Hendry, and R. Tanone, "Implementasi Model Deep Learning Convolutional Neural Network (CNN) Pada Citra Penyakit Daun Jagung Untuk Klasifikasi Penyakit Tanaman," *J. Pendidik. Teknol. Inf.*, vol. 6, no. 1, pp. 107–116, 2023.