

## Implementasi Deep Learning Menggunakan CNN Untuk Klasifikasi Tingkat Kematangan Buah Jeruk Berbasis Android

Fajar Rahardika Bahari Putra<sup>1</sup>, Muhammad Rizki Setyawan<sup>2\*</sup>, Rendra Soekarta<sup>3</sup>, Nabila<sup>4</sup>, La Jupriadi Fakhri<sup>5</sup>

<sup>1,2,3,4,5</sup>Program Studi Teknik Informatika, Fakultas Teknik, Universitas Muhammadiyah Sorong, Papua Barat Daya, 98418

Penulis Korespondensi : [rizki@um-sorong.ac.id](mailto:rizki@um-sorong.ac.id), No.Hp: 081225126513

### Article Info

Received : 25 November 2024

Revised : 16 Desember 2024

Accepted : 30 Desember 2024

**Abstract :** Indonesia is a country that heavily relies on the agricultural sector, including various types of horticultural commodities, especially fruits. One example is oranges, which have many benefits and a sweet, refreshing taste. To obtain the best flavor and freshness, fully ripe oranges are the preferred choice. However, the process of recognizing the ripeness of oranges still faces many challenges. With advances in computer technology, particularly through smartphones, many human tasks can now be performed more efficiently and practically. One useful technology is computer vision, which can be used to automatically identify and determine the ripeness of oranges. This research aims to implement Convolutional Neural Networks (CNN) to measure the model's performance and ensure its capability in classifying the ripeness of oranges. The results of the research show that classification using CNN with the VGG-16 architecture achieved a training accuracy of 96% and a validation accuracy of 97%.

**Abstrak :** Indonesia adalah negara yang sangat mengandalkan sektor pertanian, termasuk berbagai jenis komoditas hortikultura, khususnya buah-buahan. Salah satu contohnya adalah jeruk banyak manfaat dan memiliki rasa manis serta menyegarkan. Untuk mendapatkan rasa dan kesegaran terbaik, jeruk yang telah matang menjadi pilihan utama. Namun, proses mengenali tingkat kematangan buah jeruk masih menghadapi banyak tantangan. Dengan kemajuan teknologi komputer, terutama melalui perangkat smartphone, berbagai tugas manusia kini dapat dilakukan dengan lebih efisien dan praktis. Salah satu teknologi yang bermanfaat adalah computer vision, yang dapat digunakan untuk secara otomatis mengidentifikasi dan menentukan tingkat kematangan buah jeruk. Penelitian ini bertujuan mengimplementasikan *Convolutional Neural Network* (CNN) untuk mengukur performa model dan memastikan kemampuannya dalam melakukan klasifikasi kematangan buah jeruk. Hasil penelitian yang didapatkan klasifikasi menggunakan CNN dengan arsitektur VGG-16 menunjukkan tingkat akurasi pelatihan sebesar 96% dan akurasi validasi mencapai 97%.

**Keyword :** Classification, Citrus, Convolutional Neural Network, VGG 16

### PENDAHULUAN

Indonesia adalah negara dengan luas wilayah yang sangat besar, yang tercermin dari jumlah penduduk yang padat, kekayaan sumber daya alam, serta keberagaman budaya dan tradisi. Oleh karena itu, Indonesia dikenal sebagai negara dengan berbagai potensi yang melimpah. Dalam hal sumber daya alam,

Indonesia memiliki banyak keistimewaan, termasuk berbagai jenis flora dan fauna. Beragam tanaman dan buah-buahan dapat tumbuh dengan baik di berbagai wilayah di seluruh nusantara [1]. Salah satu contoh buah jeruk, yang memiliki banyak manfaat dan menjadi salah satu buah yang sangat populer di seluruh dunia. Produksi jeruk di Indonesia

pada tahun 2022 mencapai 2.551.999 ton, meningkat 13,2% dibandingkan produksi tahun sebelumnya yang sebesar 2.401.064 ton [2]. Jeruk termasuk dalam kelompok tanaman hortikultura, dan di Indonesia terdapat berbagai jenis jeruk khas dari berbagai daerah dengan ciri khas masing-masing [3]. Untuk memperoleh rasa manis dan kesegaran terbaik, jeruk yang matang dengan sempurna menjadi pilihan utama. Dalam sektor pertanian, teknologi klasifikasi citra dapat dimanfaatkan untuk berbagai tujuan, termasuk untuk menentukan tingkat kematangan buah. [4]. *Artificial Intelligence* (AI) merupakan salah satu cabang ilmu komputer dimana salah satu cabang dari AI adalah machine learning, yang kemudian berkembang lebih lanjut menjadi deep learning [5]. Saat ini, citra digital juga telah banyak digunakan untuk membantu mempermudah pekerjaan manusia. Keberadaannya tidak sulit untuk ditemukan pada objek-objek di sekitar kita. Selain itu, citra digital telah digunakan untuk alat pendeteksi penyakit hingga mendeteksi kematangan pada tanaman [6]. Setiap citra memiliki karakteristik yang berbeda, sehingga untuk mendapatkan fitur tekstur dari digunakan metode *Gray Level Co-occurrence Matrix* (GLCM). Metode ini bertujuan untuk mengekstrak fitur khas dari suatu objek, yang selanjutnya dapat digunakan dalam proses identifikasi objek tersebut. [7].

Peneliti sebelumnya [8] menghadapi masalah dalam mendeteksi tingkat kematangan buah tomat, di mana manusia memiliki keterbatasan seperti kelelahan dan kurang fokus. Hal ini sering menyebabkan hasil penyortiran yang tidak konsisten dan tidak akurat, ditambah dengan ketiadaan metode yang digunakan untuk mendeteksi kematangan buah. Peneliti [9] membahas kendala dalam produksi kopi Arabika akibat penyakit

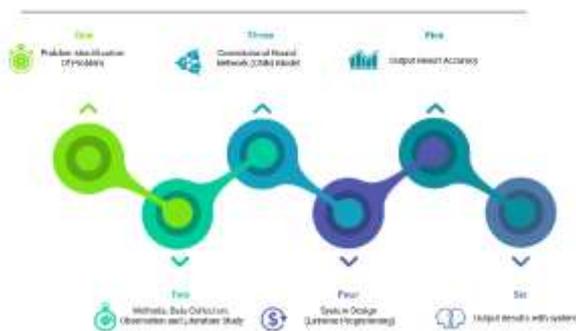
tanaman. Untuk mengatasi permasalahan ini, langkah awal yang diambil adalah mengenali penyakit pada daun kopi Arabika menggunakan metode *Convolutional Neural Network* (CNN). Peneliti [10] juga mengkaji topik serupa, yaitu mengenali penyakit tanaman pada daun kopi Arabika sebagai langkah awal pencegahan dan pengobatan infeksi guna menghindari kerusakan tanaman dengan metode CNN. Selanjutnya, peneliti [11] mengangkat permasalahan dalam pengamatan koloni bakteri yang masih dilakukan secara manual menggunakan mikroskop dengan perbesaran 1000 kali. Untuk mengatasi hal ini, digunakan kombinasi metode *Convolutional Neural Network* (CNN) dan *K-Nearest Neighbor* (KNN).

Saat ini, terdapat kesulitan dalam mengidentifikasi kematangan buah jeruk. Petani maupun konsumen sering mengalami kendala dalam memastikan kematangan buah, yang menyebabkan kesalahan dalam pemilihan buah matang dan segar. Proses manual penentuan kematangan buah jeruk memerlukan keahlian khusus dan sering kali menimbulkan perbedaan pendapat, sehingga kurang efisien untuk skala besar. Untuk mengatasi masalah ini, penelitian ini mengusulkan penggunaan metode *Convolutional Neural Network* (CNN) untuk identifikasi kematangan buah jeruk secara otomatis berbasis Android. Dengan metode ini, pengguna dapat memperoleh informasi secara cepat dan akurat hanya dengan mengambil foto buah jeruk. Berdasarkan latar belakang dan permasalahan tersebut, penulis mengangkat judul **“Implementasi Deep Learning Menggunakan Convolutional Neural Network untuk Klasifikasi Jenis Buah Jeruk Berbasis Android”** yang diharapkan dapat memberikan solusi efektif terhadap permasalahan yang ada.

## METODE PENELITIAN

### 2.1. Tahap Penelitian

Penelitian ini terdiri dari beberapa tahap [12] dan menggunakan pendekatan kuantitatif dengan metode eksperimen untuk mengevaluasi pengaruh berbagai nilai learning rate pada model deep learning *Convolutional Neural Network* (CNN) dalam mengklasifikasikan jenis buah jeruk. Eksperimen ini dirancang untuk menguji hipotesis bahwa terdapat nilai learning rate tertentu yang paling optimal dalam meningkatkan akurasi pendeteksian tingkat kematangan buah jeruk [13], [14], [15], [16]. Adapapun alur penelitian ditunjukkan pada Gambar 1 [17].



**Gambar 1 Alur Penelitian**

Penjelasan alur penelitian yang dilakukan sebagai berikut [18]:

1. Tahap awal dalam rangka penelitian. Kompilasi dan identifikasi data jenis kematangan buah jeruk dari petani kebun jeruk Bapak Sisono, Kabupaten Sorong pada tahun 2023.
2. Pengumpulan data dilakukan dengan cara studi literatur, Observasi dan wawancara langsung dengan petani jeruk mendapatkan data berupa dataset jeruk sebanyak 694.
3. Di tahap ini mengembangkan model *convolutional neural network*. Model ini digunakan untuk memproses data yang dikumpulkan dan menjalankan tugas klasifikasi.

4. Tahap ini melibatkan perancangan sistem secara menyeluruh.
5. Setelah model dijalankan, tahap ini berfokus pada evaluasi hasil. Akurasi keluaran dianalisis untuk menentukan seberapa baik model dalam melakukan klasifikasi atau pendeteksian.
6. Tahapan terakhir adalah penyajian hasil penelitian yang diperoleh dari model. Hasil ini dikaitkan dengan sistem yang dirancang untuk menunjukkan efektivitas pendekatan yang digunakan.

### 2.2. Metode *Convolutional Neural Network* (CNN)

Metode *Convolutional Neural Network* (CNN) adalah pendekatan yang dirancang untuk menganalisis data dua dimensi [19]. CNN berfungsi dalam mengenali objek dan elemen visual, serta secara luas diterapkan dalam berbagai bidang computer vision, seperti pengenalan wajah, deteksi objek, pengenalan gambar, dan analisis visual. Metode ini memungkinkan untuk mengidentifikasi objek dalam gambar maupun video. Keunggulan utama metode ini terletak pada kemampuannya mendeteksi fitur lokal pada citra melalui operasi konvolusi, sehingga dapat mengenali elemen pada citra [20].

Berikut adalah tahap pengembangan model CNN:



**Gambar 3 Pengembangan Model CNN**

Tahapan pada Gambar 3 terdiri dari: persiapan dataset, dataset preprocessing, pembuatan model, pengujian model dan pembuatan model menjadi *tensorflow lite*.



saraf, sehingga dapat meningkatkan kemampuan model untuk mengekstraksi fitur yang lebih kompleks dari data.

```
[ ] base_model.compile(optimizer=keras.optimizers.Adam, weights='imagenet', input_tensor=Input(shape=(9,9,3)),
base_model.summary())
```

**Gambar 5 Koding Pembuatan Model**

Model Gambar 8 ini menggunakan model VGG16 dari Keras sebagai model dasar untuk pembelajaran transfer dengan mengekstraksi hanya fitur-fitur dasar (feature extractor) tanpa lapisan klasifikasi akhir (include\_top=Salah). Model ini menggunakan bobot yang dilatih pada dataset ImageNet (bobot = "imagenet") dan menerima input gambar 90x90 piksel dengan 3 saluran warna (RGB), yang didefinisikan dengan input\_tensor = Input(shape = (90,90,3)). Terakhir, perintah base\_model.summary() digunakan untuk menampilkan ringkasan struktur model, membantu untuk melihat konfigurasi lapisan yang akan digunakan sebagai dasar untuk pelatihan lebih lanjut pada dataset baru.

Pada gambar 9 adalah kode yang memanggil vgg-16. Setelah mengimpor model, langkah selanjutnya adalah mengatur nilai 2 sesuai dengan kelas yang ada dan mengatur fungsi aktivasi yaitu softmax, berikut ini adalah gambar model CNN yang digunakan:

```
Model: "sequential_27"
Layer (type)                Output Shape                Param #
-----
vgg16 (Functional)          (None, 2, 2, 512)          14714688
conv2d_27 (Conv2D)         (None, 2, 2, 75)           345675
max_pooling2d_27 (MaxPoolin (None, 1, 1, 75)           0
g2D)
dropout_27 (Dropout)       (None, 1, 1, 75)           0
Flatten_27 (Flatten)       (None, 75)                  0
dense_27 (Dense)           (None, 3)                   228
-----
Total params: 15,060,591
Trainable params: 345,983
```

**Gambar 6 VGG-16**

Pada Gambar 9, Anda dapat melihat hasil keseluruhan dari 15.060.591 parameter

dan kemudian hasil dalam parameter train adalah 345.903.

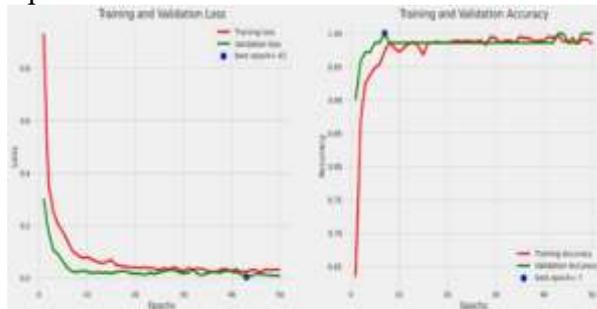
### 3.1.4. Training Dataset

Setelah melalui semua proses di atas, mulai dari memuat dataset hingga membuat model dengan menggunakan vgg16, langkah selanjutnya adalah proses pelatihan data. Proses ini menggunakan 50 epoch, berikut hasil training yang didapatkan:

```
Epoch 1/50: 100% 1/1 [0:00< 0:00] 1000/1000 0.0000 - loss: 0.0000 - acc: 0.0000 - val_loss: 0.0000 - val_acc: 0.0000
Epoch 2/50: 100% 1/1 [0:00< 0:00] 1000/1000 0.0000 - loss: 0.0000 - acc: 0.0000 - val_loss: 0.0000 - val_acc: 0.0000
Epoch 3/50: 100% 1/1 [0:00< 0:00] 1000/1000 0.0000 - loss: 0.0000 - acc: 0.0000 - val_loss: 0.0000 - val_acc: 0.0000
Epoch 4/50: 100% 1/1 [0:00< 0:00] 1000/1000 0.0000 - loss: 0.0000 - acc: 0.0000 - val_loss: 0.0000 - val_acc: 0.0000
Epoch 5/50: 100% 1/1 [0:00< 0:00] 1000/1000 0.0000 - loss: 0.0000 - acc: 0.0000 - val_loss: 0.0000 - val_acc: 0.0000
Epoch 6/50: 100% 1/1 [0:00< 0:00] 1000/1000 0.0000 - loss: 0.0000 - acc: 0.0000 - val_loss: 0.0000 - val_acc: 0.0000
Epoch 7/50: 100% 1/1 [0:00< 0:00] 1000/1000 0.0000 - loss: 0.0000 - acc: 0.0000 - val_loss: 0.0000 - val_acc: 0.0000
Epoch 8/50: 100% 1/1 [0:00< 0:00] 1000/1000 0.0000 - loss: 0.0000 - acc: 0.0000 - val_loss: 0.0000 - val_acc: 0.0000
Epoch 9/50: 100% 1/1 [0:00< 0:00] 1000/1000 0.0000 - loss: 0.0000 - acc: 0.0000 - val_loss: 0.0000 - val_acc: 0.0000
Epoch 10/50: 100% 1/1 [0:00< 0:00] 1000/1000 0.0000 - loss: 0.0000 - acc: 0.0000 - val_loss: 0.0000 - val_acc: 0.0000
Epoch 11/50: 100% 1/1 [0:00< 0:00] 1000/1000 0.0000 - loss: 0.0000 - acc: 0.0000 - val_loss: 0.0000 - val_acc: 0.0000
Epoch 12/50: 100% 1/1 [0:00< 0:00] 1000/1000 0.0000 - loss: 0.0000 - acc: 0.0000 - val_loss: 0.0000 - val_acc: 0.0000
Epoch 13/50: 100% 1/1 [0:00< 0:00] 1000/1000 0.0000 - loss: 0.0000 - acc: 0.0000 - val_loss: 0.0000 - val_acc: 0.0000
Epoch 14/50: 100% 1/1 [0:00< 0:00] 1000/1000 0.0000 - loss: 0.0000 - acc: 0.0000 - val_loss: 0.0000 - val_acc: 0.0000
Epoch 15/50: 100% 1/1 [0:00< 0:00] 1000/1000 0.0000 - loss: 0.0000 - acc: 0.0000 - val_loss: 0.0000 - val_acc: 0.0000
Epoch 16/50: 100% 1/1 [0:00< 0:00] 1000/1000 0.0000 - loss: 0.0000 - acc: 0.0000 - val_loss: 0.0000 - val_acc: 0.0000
Epoch 17/50: 100% 1/1 [0:00< 0:00] 1000/1000 0.0000 - loss: 0.0000 - acc: 0.0000 - val_loss: 0.0000 - val_acc: 0.0000
Epoch 18/50: 100% 1/1 [0:00< 0:00] 1000/1000 0.0000 - loss: 0.0000 - acc: 0.0000 - val_loss: 0.0000 - val_acc: 0.0000
Epoch 19/50: 100% 1/1 [0:00< 0:00] 1000/1000 0.0000 - loss: 0.0000 - acc: 0.0000 - val_loss: 0.0000 - val_acc: 0.0000
Epoch 20/50: 100% 1/1 [0:00< 0:00] 1000/1000 0.0000 - loss: 0.0000 - acc: 0.0000 - val_loss: 0.0000 - val_acc: 0.0000
Epoch 21/50: 100% 1/1 [0:00< 0:00] 1000/1000 0.0000 - loss: 0.0000 - acc: 0.0000 - val_loss: 0.0000 - val_acc: 0.0000
Epoch 22/50: 100% 1/1 [0:00< 0:00] 1000/1000 0.0000 - loss: 0.0000 - acc: 0.0000 - val_loss: 0.0000 - val_acc: 0.0000
Epoch 23/50: 100% 1/1 [0:00< 0:00] 1000/1000 0.0000 - loss: 0.0000 - acc: 0.0000 - val_loss: 0.0000 - val_acc: 0.0000
Epoch 24/50: 100% 1/1 [0:00< 0:00] 1000/1000 0.0000 - loss: 0.0000 - acc: 0.0000 - val_loss: 0.0000 - val_acc: 0.0000
Epoch 25/50: 100% 1/1 [0:00< 0:00] 1000/1000 0.0000 - loss: 0.0000 - acc: 0.0000 - val_loss: 0.0000 - val_acc: 0.0000
Epoch 26/50: 100% 1/1 [0:00< 0:00] 1000/1000 0.0000 - loss: 0.0000 - acc: 0.0000 - val_loss: 0.0000 - val_acc: 0.0000
Epoch 27/50: 100% 1/1 [0:00< 0:00] 1000/1000 0.0000 - loss: 0.0000 - acc: 0.0000 - val_loss: 0.0000 - val_acc: 0.0000
Epoch 28/50: 100% 1/1 [0:00< 0:00] 1000/1000 0.0000 - loss: 0.0000 - acc: 0.0000 - val_loss: 0.0000 - val_acc: 0.0000
Epoch 29/50: 100% 1/1 [0:00< 0:00] 1000/1000 0.0000 - loss: 0.0000 - acc: 0.0000 - val_loss: 0.0000 - val_acc: 0.0000
Epoch 30/50: 100% 1/1 [0:00< 0:00] 1000/1000 0.0000 - loss: 0.0000 - acc: 0.0000 - val_loss: 0.0000 - val_acc: 0.0000
Epoch 31/50: 100% 1/1 [0:00< 0:00] 1000/1000 0.0000 - loss: 0.0000 - acc: 0.0000 - val_loss: 0.0000 - val_acc: 0.0000
Epoch 32/50: 100% 1/1 [0:00< 0:00] 1000/1000 0.0000 - loss: 0.0000 - acc: 0.0000 - val_loss: 0.0000 - val_acc: 0.0000
Epoch 33/50: 100% 1/1 [0:00< 0:00] 1000/1000 0.0000 - loss: 0.0000 - acc: 0.0000 - val_loss: 0.0000 - val_acc: 0.0000
Epoch 34/50: 100% 1/1 [0:00< 0:00] 1000/1000 0.0000 - loss: 0.0000 - acc: 0.0000 - val_loss: 0.0000 - val_acc: 0.0000
Epoch 35/50: 100% 1/1 [0:00< 0:00] 1000/1000 0.0000 - loss: 0.0000 - acc: 0.0000 - val_loss: 0.0000 - val_acc: 0.0000
Epoch 36/50: 100% 1/1 [0:00< 0:00] 1000/1000 0.0000 - loss: 0.0000 - acc: 0.0000 - val_loss: 0.0000 - val_acc: 0.0000
Epoch 37/50: 100% 1/1 [0:00< 0:00] 1000/1000 0.0000 - loss: 0.0000 - acc: 0.0000 - val_loss: 0.0000 - val_acc: 0.0000
Epoch 38/50: 100% 1/1 [0:00< 0:00] 1000/1000 0.0000 - loss: 0.0000 - acc: 0.0000 - val_loss: 0.0000 - val_acc: 0.0000
Epoch 39/50: 100% 1/1 [0:00< 0:00] 1000/1000 0.0000 - loss: 0.0000 - acc: 0.0000 - val_loss: 0.0000 - val_acc: 0.0000
Epoch 40/50: 100% 1/1 [0:00< 0:00] 1000/1000 0.0000 - loss: 0.0000 - acc: 0.0000 - val_loss: 0.0000 - val_acc: 0.0000
Epoch 41/50: 100% 1/1 [0:00< 0:00] 1000/1000 0.0000 - loss: 0.0000 - acc: 0.0000 - val_loss: 0.0000 - val_acc: 0.0000
Epoch 42/50: 100% 1/1 [0:00< 0:00] 1000/1000 0.0000 - loss: 0.0000 - acc: 0.0000 - val_loss: 0.0000 - val_acc: 0.0000
Epoch 43/50: 100% 1/1 [0:00< 0:00] 1000/1000 0.0000 - loss: 0.0000 - acc: 0.0000 - val_loss: 0.0000 - val_acc: 0.0000
Epoch 44/50: 100% 1/1 [0:00< 0:00] 1000/1000 0.0000 - loss: 0.0000 - acc: 0.0000 - val_loss: 0.0000 - val_acc: 0.0000
Epoch 45/50: 100% 1/1 [0:00< 0:00] 1000/1000 0.0000 - loss: 0.0000 - acc: 0.0000 - val_loss: 0.0000 - val_acc: 0.0000
Epoch 46/50: 100% 1/1 [0:00< 0:00] 1000/1000 0.0000 - loss: 0.0000 - acc: 0.0000 - val_loss: 0.0000 - val_acc: 0.0000
Epoch 47/50: 100% 1/1 [0:00< 0:00] 1000/1000 0.0000 - loss: 0.0000 - acc: 0.0000 - val_loss: 0.0000 - val_acc: 0.0000
Epoch 48/50: 100% 1/1 [0:00< 0:00] 1000/1000 0.0000 - loss: 0.0000 - acc: 0.0000 - val_loss: 0.0000 - val_acc: 0.0000
Epoch 49/50: 100% 1/1 [0:00< 0:00] 1000/1000 0.0000 - loss: 0.0000 - acc: 0.0000 - val_loss: 0.0000 - val_acc: 0.0000
Epoch 50/50: 100% 1/1 [0:00< 0:00] 1000/1000 0.0000 - loss: 0.0000 - acc: 0.0000 - val_loss: 0.0000 - val_acc: 0.0000
```

**Gambar 7 Training Dataset**

Berdasarkan Gambar 10, akurasi model pada pelatihan mencapai 98%, sementara akurasi validasi mencapai 97%. Proses pelatihan dilakukan menggunakan gambar dengan ukuran input 32 x 32 px. Waktu yang dibutuhkan untuk melatih model selama 50 epoch adalah 15 menit. Semakin banyak dataset yang digunakan dan semakin banyak jumlah epoch, semakin lama waktu yang diperlukan untuk melatih model.



**Gambar 8 Hasil Grafik Training**

Grafik pada Gambar 11 ini menunjukkan performa model selama pelatihan melalui metrik Loss dan Accuracy untuk data pelatihan dan validasi selama 50 epoch.

1. Grafik Kiri (Loss Pelatihan dan Validasi): Grafik ini menggambarkan penurunan loss (error) pada data training (garis merah) dan data validasi (garis hijau). Pada epoch awal, kerugian pelatihan turun tajam yang

mengindikasikan model sedang belajar, dan setelah sekitar 10 epoch, kerugian menjadi stabil mendekati nol. Titik biru pada epoch 43 menunjukkan epoch terbaik berdasarkan kerugian validasi terendah.

2. Grafik Kanan (Akurasi Pelatihan dan Validasi): Grafik ini menunjukkan peningkatan akurasi pada data pelatihan (garis merah) dan validasi (garis hijau). Awalnya, akurasi meningkat tajam dan mendekati 100% di sekitar epoch ke-7, di mana titik biru menunjukkan epoch terbaik dengan akurasi validasi tertinggi.

Secara keseluruhan, grafik ini menunjukkan bahwa model telah mencapai kinerja yang baik dengan kehilangan konvergensi yang rendah dan akurasi yang tinggi, yang mengindikasikan bahwa model mampu mengenali data training dan validasi dengan sangat baik.

### 3.1.5. Confusion Matriks

Hasil dari pengujian ini didapatkan nilai TP (Benar Positive) dengan nilai 15, TN (Benar Negative) dengan nilai 18, FP (Salah Positive) dengan nilai 18, dan FN (Salah Negative) dengan nilai 18. Untuk lebih jelasnya akan ditunjukkan pada gambar di bawah ini:



**Gambar 9 Confusion Matriks**

Pada Gambar 13, Confusion Matrix menunjukkan hasil klasifikasi model pada tiga

kelas yaitu non-jeruk, matang, dan mentah. Baris pada confusion matrix menunjukkan kelas aktual, sedangkan kolom menunjukkan kelas prediksi. Untuk kelas non-oranges, terdapat 7 contoh yang diklasifikasikan dengan benar sebagai non-oranges (Benar positive), dan tidak ada kesalahan klasifikasi (Salah positive atau Salah negative) untuk kelas ini. Untuk kelas matang, terdapat 22 contoh yang diklasifikasikan dengan benar sebagai matang, tetapi ada 3 contoh yang salah diklasifikasikan sebagai mentah. Untuk kelas mentah, ada 38 contoh yang diklasifikasikan dengan benar sebagai mentah, dan tidak ada contoh yang salah diklasifikasikan sebagai mentah.

	precision	recall	f1-score	support
bukanjeruk	1.00	1.00	1.00	7
matang	1.00	0.88	0.94	25
mentah	0.93	1.00	0.96	38
accuracy			0.96	70
macro avg	0.98	0.96	0.97	70
weighted avg	0.96	0.96	0.96	70

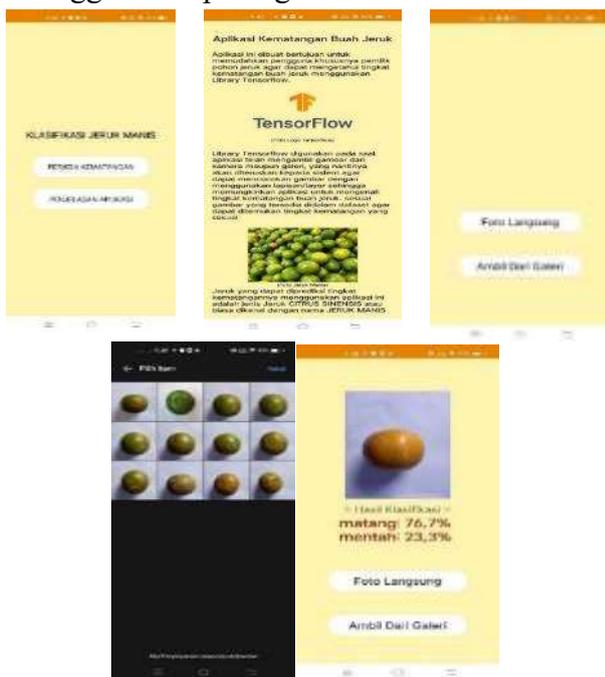
**Gambar 10 Hasil Matriks**

Gambar 14 Laporan Klasifikasi, yang menyajikan metrik evaluasi untuk setiap kelas dalam bentuk presisi, recall, f1-score, dan support. Presisi menunjukkan proporsi prediksi yang benar dari semua prediksi untuk kelas tertentu. Semua kelas memiliki presisi yang tinggi (1.00 untuk non-matang dan matang, dan 0.93 untuk mentah). Recall menunjukkan proporsi contoh dari kelas tertentu yang diklasifikasikan dengan benar. Recall untuk non-jeruk dan mentah adalah 1,00 (semua contoh diklasifikasikan dengan benar), sedangkan untuk matang adalah 0,88 (karena ada 3 contoh yang salah diklasifikasikan). F1-Score adalah rata-rata harmonis dari presisi dan recall, yang menunjukkan keseimbangan antara keduanya. F1-Score untuk non-jeruk adalah 1.00, untuk jeruk matang adalah 0.94, dan untuk jeruk

mentah adalah 0.96. Dukungan menunjukkan jumlah total contoh untuk setiap kelas dalam dataset. Akurasi keseluruhan model adalah 0,96 atau 96%, dengan rata-rata F1-Score makro 0,97 dan rata-rata F1-Score tertimbang 0,96, yang menunjukkan kinerja model yang sangat baik.

### 3.1.6. Implementasi Aplikasi Android

Aplikasi android dibuat dengan menggunakan software Android Studio, dengan menggunakan bahasa python dan kode script Java dalam penulisan codingnya. Untuk menghubungkan bahasa pemrograman Python dan kode script Java dalam pembuatan android menggunakan android pada Bahasa Python. Memberikan beberapa gambar pada tabel yang merupakan hasil deteksi/klasifikasi menggunakan perangkat lunak:



Gambar 11 Hasil Klasifikasi Kematangan

### 3.1.7. Pengujian Model

Pada tahap pengujian ini, peneliti menggunakan 10 citra yang terdiri dari masing-masing kelas. Berikut ini adalah tabel hasil pengujian model yang dilakukan. Berikut hasil prediksi yang didapatkan dengan metode CNN, ditunjukkan pada tabel 2 berikut:

Tabel 1 Hasil Prediksi Kematang Buah

Label	Gambar	Prediksi	Hasil
Mentah	 = Hasil Klasifikasi = matang: 60,1% mentah: 39,9%	Matang	Salah
Matang	 = Hasil Klasifikasi = matang: 91,1% mentah: 8,9%	Matang	Benar
Mentah	 = Hasil Klasifikasi = matang: 31,5% mentah: 68,5%	Mentah	Benar
Matang	 = Hasil Klasifikasi = matang: 66,0% mentah: 34,0%	Mentah	Benar
Mentah	 = Hasil Klasifikasi = matang: 72,5% mentah: 27,4%	Matang	Salah
Matang	 = Hasil Klasifikasi = matang: 42,6% mentah: 57,4%	Mentah	Salah
Matang	 = Hasil Klasifikasi = matang: 26,3% mentah: 73,7%	Mentah	Salah
Mentah	 = Hasil Klasifikasi = matang: 44,7% mentah: 55,3%	Mentah	Benar
Matang	 = Hasil Klasifikasi = matang: 42,6% mentah: 57,4%	Mentah	Salah

Matang



Matang Benar

Hasil pengujian dikatakan benar jika label data yang diuji sesuai dengan prediksi model. Pada tabel di atas, dapat dilihat bahwa hasil pengujian 10 data gambar mendapatkan hasil 5 pengujian mendapatkan hasil benar dan 5 pengujian salah dalam melakukan prediksi.

### KESIMPULAN DAN SARAN

Berdasarkan penelitian yang telah dilakukan, dapat disimpulkan bahwa proses klasifikasi kematangan buah jeruk telah berhasil diimplementasikan dengan baik, menunjukkan bahwa aplikasi Android yang dibangun menggunakan Android Studio dapat berjalan sesuai dengan tahapan yang telah direncanakan. Untuk memastikan sistem berfungsi dengan baik, dilakukan pengujian black box dengan skenario pengujian untuk memastikan bahwa aplikasi berjalan sesuai dengan fungsinya. Hasil klasifikasi kematangan buah jeruk menggunakan Convolutional Neural Network (CNN) dengan arsitektur VGG16 menunjukkan tingkat akurasi pelatihan sebesar 90% dan akurasi validasi sebesar 92%.

Saran yang dapat diberikan untuk pengembangan sistem klasifikasi kematangan buah jeruk ini adalah melakukan beberapa perbaikan agar hasilnya lebih optimal, seperti menambah jumlah kelas dataset agar lebih bervariasi dalam pengklasifikasian, serta memungkinkan klasifikasi dilakukan secara real-time. Saat ini, sistem hanya dapat berjalan di Android, namun diharapkan ke depannya dapat diimplementasikan juga di sistem iOS. Selain itu, penambahan dataset terkait tekstur kulit jeruk juga bisa meningkatkan kinerja model.

### DAFTAR PUSTAKA

- [1] Y. T. L. Calvin Bill Roring, Dadang Iskandar Mulyana, “Klasifikasi Tingkat Kematangan Buah Jambu Bol Berdasarkan Warna Kulit Menggunakan Metode Naïve Bayes,” *J. Pendidik. Tambusai*, vol. 6, no. 1, pp. 2938–2948, 2022.
- [2] J. C. Lapendy, A. A. C. Resky, H. Makmur, A. B. Kaswar, D. D. Andayani, and F. Adiba, “Klasifikasi Rasa Jeruk Siam Berdasarkan Warna Dan Tekstur Berbasis Pengolahan Citra Digital,” *JIPi (Jurnal Ilm. Penelit. dan Pembelajaran Inform.)*, vol. 9, no. 2, pp. 756–767, 2024, doi: dx.doi.org/10.26418/positron.v12i2.64594
- [3] A. M. Simarmata, A. Zizwan Putra, and A. Mahmud Husein, “Penerapan Metode Computer Vision Dalam Klasifikasi Buah Jeruk Menggunakan Teknik Image Pre-Processing,” *Data Sci. Indones.*, vol. 3, no. 2, pp. 110–116, 2023, doi: 10.47709/dsi.v3i2.4010.
- [4] N. M. Dwicahyo, R. Wulanningrum, Aswi, and R. Ramadhani, “Penyortiran Buah Jeruk Dengan Ekraksi Ciri Rgb To Hsv Menggunakan Naïve Bayes,” *INOTEK*, vol. 8, pp. 894–902, 2024, doi: doi.org/10.29407/inotek.v8i2.5018.
- [5] R. Soekarta, N. Nurdjan, and A. Syah, “Klasifikasi Penyakit Tanaman Tomat

- Menggunakan Metode Convolutional Neural Network (CNN),” *Insect (Informatics Secur. J. Tek. Inform.*, vol. 8, no. 2, pp. 143–151, 2023, doi: doi.org/10.33506/insect.v8i2.2356.
- [6] A. S. J. Putra, I. M. I. Subroto, and B. S. W. Poetro, “Identifikasi Kematangan Buah Jeruk Medan Menggunakan K-Nearest Neighbor berbasis Metrik RGB,” *J. Transistor Elektro dan Inform. (TRANSISTOR EI)*, vol. 5, no. 3, p. 50112, 2023, doi: dx.doi.org/10.30659/ei.5.3.155-160.
- [7] Y. Sutanto, C. P. H., Nurhasanah, D. Wahyuni, Y. Arman, and Hasanuddin, “Implementasi Convolutional Neural Network dalam Menentukan Tingkat Kematangan Jeruk Siam Pontianak Berdasarkan Citra,” *Positron*, vol. 12, no. 2, pp. 163–170, 2022, doi: dx.doi.org/10.26418/positron.v12i2.64594 .
- [8] S. Sanjaya, “Aplikasi Pengenalan Tingkat Kematangan Buah Tomat Menggunakan Fitur Warna Hsv Berbasis Android,” *J. Teknoinfo*, vol. 16, no. 1, p. 26, 2022, doi: doi.org/10.33365/jti.v16i1.1489.
- [9] Yazid AUFAR, Muhammad Helmy Abdillah, and Jiki Romadoni, “Web-based CNN Application for Arabica Coffee Leaf Disease Prediction in Smart Agriculture,” *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 7, no. 1, pp. 71–79, 2023, doi: doi.org/10.29207/resti.v7i1.4622.
- [10] A. R. I. Fauzy and Erwin Budi Setiawan, “Detecting Fake News on Social Media Combined with the CNN Methods,” *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 7, no. 2, pp. 271–277, 2023, doi: doi.org/10.29207/resti.v7i2.4889.
- [11] W. N. Waluyo, R. Rizal Isnanto, and Adian Fatchur Rochim, “Comparison of Mycobacterium Tuberculosis Image Detection Accuracy Using CNN and Combination CNN-KNN,” *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 7, no. 1, pp. 80–87, 2023, doi: doi.org/10.29207/resti.v7i1.4626.
- [12] Jalu Nusantoro, Faldo Fajri Afrinanto, Wana Salam Labibah, Zamah Sari, and Yufis Azhar, “Detection of Covid-19 on X-Ray Image of Human Chest Using CNN and Transfer Learning,” *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 6, no. 3, pp. 430–441, 2022, doi: 10.29207/resti.v6i3.4118.
- [13] E. G. Sihombing, E. Aristawati, and L. S. Dewi, “Application of Learning Rate in Artificial Neural Networks to Increase Prediction Accuracy on Rubber Tree Maintenance Costs Universitas Bina Sarana Informatika , Indonesia Universitas Nusa Mandiri , Jakarta ,

- Indonesia,” *KESATRIA J. Penerapan Sist. Inf. (Komputer Manajemen)*, vol. 5, no. 4, pp. 1461–1470, 2024, doi: doi.org/10.30645/kesatria.v5i4.467.g462.
- [14] H. Nurrani, Andi Kurniawan Nugroho, and Sri Heranurweni, “Image Classification of Vegetable Quality using Support Vector Machine based on Convolutional Neural Network,” *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 7, no. 1, pp. 168–178, 2023, doi: doi.org/10.29207/resti.v7i1.4715.
- [15] Mayanda Mega Santoni, Nurul Chamidah, Desta Sandya Prasvita, Helena Nurramdhani Irmanda, Ria Astriratma, and Reza Amarta Prayoga, “Penerapan Convolutional Neural Networks untuk Mesin Penerjemah Bahasa Daerah Minangkabau Berbasis Gambar,” *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 5, no. 6, pp. 1153–1160, 2021, doi: doi.org/10.29207/resti.v5i6.3614.
- [16] Rima Dias Ramadhani, A. Nur Aziz Thohari, C. Kartiko, A. Junaidi, T. Ginanjar Laksana, and N. Alim Setya Nugraha, “Optimasi Akurasi Metode Convolutional Neural Network untuk Identifikasi Jenis Sampah,” *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 5, no. 2, pp. 312–318, 2021, doi: doi.org/10.29207/resti.v5i2.2754.
- [17] R. Sistem *et al.*, “Analisis Perbandingan Algoritma Klasifikasi MLP dan CNN pada Dataset American Sign Language,” *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 1, no. 10, pp. 489–495, 2021, doi: doi.org/10.29207/resti.v5i3.3009.
- [18] F. R. B. Putra, A. Fadlil, and R. Umar, “Application of Forward Chaining Method , Certainty Factor , and Bayes Theorem for Cattle Disease,” *Int. J. Adv. Sci. Eng. Inf. Technol.*, vol. 14, no. 1, pp. 365–374, 2024, doi: https://doi.org/10.18517/ijaseit.14.1.18912 .
- [19] I. Salamah, S. Humairoh, and S. Soim, “Implementasi Convolutional Neural Network Pada Alat Klasifikasi Kematangan dan Ukuran Buah Nanas Berbasis Android,” *INOVTEK Polbeng - Seri Inform.*, vol. 8, no. 2, p. 243, 2023, doi: doi.org/10.35314/isi.v8i2.3413.
- [20] J. Christian and S. I. Al Idrus, “Introduction to Citrus Fruit Ripens Using the Deep Learning Convolutional Neural Network (CNN) Learning Method,” *Asian J. Appl. Educ.*, vol. 2, no. 3, pp. 459–470, 2023, doi: doi.org/10.55927/ajae.v2i3.5003.