

Identifikasi Jenis Jambu Biji Real-Time Menggunakan Convolutional Neural Network

Rino Syahra Prasetyo¹

^{1,2}Teknik Informatika, Fakultas Teknik dan Ilmu Komputer, Universitas Nusantara PGRI Kediri

E-mail: [*¹rinasyahra3502@gmail.com](mailto:rinasyahra3502@gmail.com).

Abstrak – Penelitian ini membahas pengembangan sistem identifikasi jenis jambu biji secara real-time menggunakan metode Convolutional Neural Network (CNN). Pemilihan topik ini dilatarbelakangi oleh masih banyaknya proses identifikasi jenis jambu biji yang dilakukan secara manual sehingga berpotensi menimbulkan kesalahan klasifikasi. Metode penelitian yang digunakan adalah pendekatan deep learning dengan CNN melalui tahapan pengumpulan dataset citra jambu biji merah dan jambu biji putih, pra-pemrosesan citra, pelatihan model, serta pengujian dan evaluasi sistem. Model CNN dilatih menggunakan data citra digital dan diimplementasikan pada sistem berbasis kamera untuk melakukan identifikasi secara langsung. Hasil pengujian menunjukkan bahwa sistem mampu mengklasifikasikan jenis jambu biji dengan tingkat akurasi yang tinggi serta waktu respon yang cepat. Evaluasi menggunakan confusion matrix memperlihatkan nilai presisi, recall, dan F1-score yang seimbang pada masing-masing kelas. Dengan demikian, penelitian ini membuktikan bahwa penerapan CNN efektif digunakan untuk identifikasi jenis jambu biji secara real-time dan berpotensi dikembangkan lebih lanjut sebagai sistem pendukung di bidang pertanian dan pengolahan hasil buah.

Kata Kunci — Convolutional Neural Network, deep learning, identifikasi citra, jambu biji, real-time

1. PENDAHULUAN

Jambu biji merupakan salah satu tanaman buah tropis yang banyak dibudidayakan di Indonesia dan memiliki nilai ekonomi yang cukup tinggi. Tanaman ini memiliki beberapa varietas dengan karakteristik yang berbeda, khususnya pada warna daging buah, seperti jambu biji merah dan jambu biji putih. Perbedaan jenis tersebut berpengaruh terhadap preferensi konsumen, proses pengolahan, serta nilai jual di pasar. Oleh karena itu, identifikasi jenis jambu biji menjadi aspek penting dalam kegiatan budidaya, distribusi, dan pengolahan hasil pertanian [1].

Pada praktiknya, proses identifikasi jenis jambu biji masih banyak dilakukan secara manual dengan mengandalkan pengamatan visual oleh petani atau pelaku usaha. Metode ini sangat bergantung pada pengalaman dan subjektivitas manusia, sehingga berpotensi menimbulkan kesalahan klasifikasi, terutama ketika kondisi pencahayaan kurang baik atau karakteristik fisik buah mengalami perubahan akibat proses penyimpanan. Kesalahan dalam identifikasi jenis buah dapat berdampak pada penurunan kualitas produk serta ketidaksesuaian dengan kebutuhan pasar. Oleh sebab itu, diperlukan suatu sistem otomatis yang mampu melakukan identifikasi jenis jambu biji secara cepat dan akurat [2].

Perkembangan teknologi pengolahan citra digital dan kecerdasan buatan membuka peluang untuk mengatasi permasalahan tersebut. Salah satu metode yang banyak digunakan dalam klasifikasi citra adalah *Convolutional Neural Network* (CNN). CNN merupakan bagian dari *deep learning* yang memiliki kemampuan untuk mengekstraksi fitur citra secara otomatis melalui proses konvolusi dan *pooling*. Metode ini telah terbukti mampu memberikan kinerja yang baik dalam berbagai permasalahan pengenalan pola dan klasifikasi objek berbasis citra digital [3].

Beberapa penelitian sebelumnya menunjukkan bahwa CNN sangat efektif dalam bidang pertanian, khususnya dalam klasifikasi citra tanaman dan buah. menunjukkan bahwa CNN mampu mengklasifikasikan citra tanaman dengan tingkat akurasi yang tinggi [4]. Penelitian lain juga membuktikan bahwa CNN dapat digunakan untuk mengenali berbagai jenis penyakit daun dan objek pertanian dengan hasil yang cukup baik [5]. Selain itu, pemanfaatan CNN dalam sistem *real-time* berbasis kamera memungkinkan proses identifikasi dilakukan secara langsung tanpa memerlukan proses pasca-pengolahan yang kompleks [6].

Berdasarkan penelitian-penelitian tersebut, dapat disimpulkan bahwa CNN memiliki potensi besar untuk diterapkan dalam sistem identifikasi jenis jambu biji. Dengan mengombinasikan CNN dan pemrosesan citra *real-time*, diharapkan sistem yang dibangun mampu memberikan hasil identifikasi secara cepat, akurat, dan mudah digunakan. Oleh karena itu, penelitian ini bertujuan untuk mengembangkan sistem identifikasi jenis jambu biji

secara real-time menggunakan metode *Convolutional Neural Network* sebagai solusi otomatis dalam proses klasifikasi buah berbasis citra digital.

2. METODE PENELITIAN

Metode penelitian yang digunakan dalam penelitian ini adalah metode eksperimental dengan pendekatan *deep learning* menggunakan *Convolutional Neural Network (CNN)*. Tahapan penelitian dirancang secara sistematis mulai dari pengumpulan *dataset*, pra-pemrosesan data, pelatihan *model*, hingga pengujian dan evaluasi sistem. Metode ini bertujuan untuk menghasilkan sistem yang mampu mengidentifikasi jenis jambu biji secara *real-time* berdasarkan citra digital [7].

2.1 Dataset Jambu Biji

Dataset yang digunakan dalam penelitian ini berupa citra digital buah jambu biji yang terdiri dari dua kelas, yaitu jambu biji merah dan jambu biji putih. Data diperoleh secara mandiri melalui proses pengambilan gambar menggunakan kamera digital dan *webcam* dengan variasi sudut pengambilan, jarak, serta kondisiahayaan yang berbeda. Seluruh citra disimpan dalam format digital dan dikelompokkan berdasarkan kelas masing-masing. *Dataset* kemudian dibagi menjadi data latih (*training data*) dan data validasi (*validation data*) dengan perbandingan tertentu untuk memastikan model dapat melakukan generalisasi dengan baik terhadap data baru yang belum pernah dilihat sebelumnya.

2.2 Convolutional Neural Network

Tahap pra-pemrosesan data dilakukan untuk menyesuaikan citra *input* dengan kebutuhan *model CNN*. Proses ini meliputi perubahan ukuran citra menjadi resolusi tertentu, normalisasi nilai piksel ke rentang 0–1, serta penyesuaian format warna citra. Selain itu, dilakukan augmentasi data untuk meningkatkan keragaman *dataset*, seperti rotasi, pergeseran posisi, dan pembalikan citra secara horizontal. Pra-pemrosesan ini bertujuan untuk meningkatkan kemampuan model dalam mengenali pola citra jambu biji pada berbagai kondisi [8].

2.3 Confusion Matrix

Confusion matrix digunakan sebagai alat evaluasi untuk mengukur kinerja model klasifikasi [9]. *Confusion matrix* menampilkan perbandingan antara hasil prediksi model dengan label sebenarnya dari data uji. Berdasarkan *confusion matrix*, beberapa parameter evaluasi dapat dihitung, antara lain akurasi (*accuracy*), presisi (*precision*), recall, dan F1-score. Parameter-parameter tersebut digunakan untuk mengetahui tingkat keberhasilan *model CNN* dalam mengidentifikasi jenis jambu biji. Persamaan yang digunakan dalam perhitungan evaluasi ditunjukkan sebagai berikut:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \dots\dots\dots(1)$$

$$Recall = \frac{TP}{TP+FN} \dots\dots\dots(2)$$

$$Precision = \frac{TP}{TP+FP} \dots\dots\dots(3)$$

$$F1 = \frac{2 \times Recall \times Precision}{Recall + Precision} \dots\dots\dots(4)$$

2.4 Alur Tahapan Penelitian

Alur tahapan penelitian menggambarkan proses yang dilakukan secara sistematis dalam penelitian ini. Tahapan penelitian dimulai dengan pengumpulan *dataset* citra jambu biji yang akan digunakan sebagai data latih dan data uji. Selanjutnya, dilakukan pra-pemrosesan citra untuk menyesuaikan data dengan kebutuhan *model CNN*. Setelah itu, *model CNN* dilatih menggunakan data latih yang telah diproses. Tahap berikutnya adalah pengujian model menggunakan data uji untuk memperoleh hasil klasifikasi. Pada tahap akhir, dilakukan evaluasi terhadap hasil pengujian menggunakan *confusion matrix* untuk mengetahui tingkat akurasi dan performa sistem secara keseluruhan [10].



Gambar 1. Alur tahapan penelitian

3. HASIL DAN PEMBAHASAN

3.1 Alur Tahapan Penelitian

Pembahasan Dalam penelitian ini, proses implementasi sistem dilakukan menggunakan perangkat lunak *Visual Studio Code* dengan dukungan bahasa pemrograman *Python*. Beberapa *library* yang digunakan antara lain *TensorFlow* dan *Keras* untuk membangun model *CNN*, *OpenCV* untuk pengolahan citra, *NumPy* untuk pengolahan data numerik, serta *Scikit-learn* untuk proses evaluasi model. Tahapan implementasi sistem yang dilakukan pada penelitian ini dijelaskan sebagai berikut.

1. Import Library

Tahap awal implementasi adalah melakukan *import* terhadap seluruh *library* yang dibutuhkan. *Library TensorFlow* dan *Keras* digunakan untuk membangun dan melatih model *CNN*, sedangkan *OpenCV* digunakan untuk membaca dan memproses citra digital. *Library* tambahan seperti *NumPy* dan *Scikit-learn* digunakan untuk mendukung pengolahan data dan evaluasi hasil klasifikasi. Proses *import library* ini merupakan tahapan dasar agar seluruh fungsi yang dibutuhkan dapat digunakan pada tahap selanjutnya.

```
import tensorflow as tf
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras import layers, models
from tensorflow.keras.callbacks import EarlyStopping
import matplotlib.pyplot as plt
import json
```

Gambar 2. Proses *Import Library*

2. Pemanggilan Dataset dan Pra-pemrosesan

Dataset citra jambu biji yang tersimpan pada direktori lokal dipanggil ke dalam sistem. Seluruh citra kemudian melalui tahap *pra-pemrosesan* yang meliputi perubahan ukuran citra menjadi resolusi seragam serta normalisasi nilai piksel. Tahap ini bertujuan untuk mempercepat proses komputasi serta memastikan keseragaman *input* pada model *CNN*. Hasil *pra-pemrosesan* citra dapat ditampilkan dalam bentuk gambar untuk menunjukkan perbedaan sebelum dan sesudah pemrosesan.

```
datagen = ImageDataGenerator(
    rescale=1./255,
    rotation_range=15,
    width_shift_range=0.1,
    height_shift_range=0.1,
    horizontal_flip=True,
    validation_split=0.2
)
```

Gambar 3. Proses *preprocessing*

3. Perubahan Label Jadi Angka

Pada tahap ini, *label* kelas yang awalnya berupa teks diubah menjadi nilai numerik. Proses pengubahan *label* ini bertujuan untuk memudahkan model *CNN* dalam melakukan proses pembelajaran. Setiap kelas jambu biji diberikan *label* numerik yang berbeda sehingga model dapat mengenali perbedaan antar kelas secara sistematis.

4. Pembagian Dataset

Dataset yang telah diproses kemudian dibagi menjadi data latih dan data uji. Data latih digunakan untuk proses pelatihan model *CNN*, sedangkan data uji digunakan untuk mengukur kemampuan model dalam mengklasifikasikan data yang belum pernah dilihat sebelumnya. Pembagian *dataset* ini dilakukan dengan perbandingan tertentu agar proses pelatihan dan pengujian berjalan seimbang

5. Pembangunan Arsitektur CNN

Arsitektur *CNN* dibangun dengan beberapa lapisan utama, yaitu *convolution layer*, *pooling layer*, dan *fully connected layer*. Lapisan konvolusi berfungsi untuk mengekstraksi fitur citra jambu biji, sedangkan lapisan pooling digunakan untuk mengurangi dimensi fitur. Lapisan *fully connected* berperan dalam menentukan kelas output berdasarkan fitur yang telah diekstraksi. Arsitektur model *CNN* yang digunakan dapat digambarkan dalam bentuk diagram arsitektur jaringan.

```
model = models.Sequential([
    layers.Conv2D(32, (3,3), activation='relu', input_shape=(IMG, IMG, 3)),
    layers.MaxPooling2D(2,2),

    layers.Conv2D(64, (3,3), activation='relu'),
    layers.MaxPooling2D(2,2),

    layers.Conv2D(128, (3,3), activation='relu'),
    layers.MaxPooling2D(2,2),

    layers.Flatten(),
    layers.Dense(128, activation='relu'),
    layers.Dropout(0.3),
    layers.Dense(train.num_classes, activation='softmax')
])
```

Gambar 4. Pembangunan modelCNN

6. Proses Training Model

Setelah arsitektur *CNN* dibangun, proses pelatihan model dilakukan menggunakan data latih. Proses *training* dilakukan dalam beberapa *epoch* dengan memantau nilai *loss* dan *accuracy* pada setiap iterasi. Selain itu, data validasi digunakan untuk melihat *performa* model selama proses pelatihan. Grafik hasil pelatihan berupa grafik akurasi dan *loss* dapat digunakan untuk menunjukkan perkembangan *performa* model.

7. Evaluasi Model

Evaluasi model dilakukan untuk mengetahui tingkat keberhasilan klasifikasi yang dihasilkan oleh *CNN*. Parameter evaluasi yang digunakan meliputi akurasi, presisi, *recall*, dan *F1-score*. Evaluasi ini dilakukan menggunakan *confusion matrix* yang menunjukkan perbandingan antara hasil prediksi dan data aktual. Berdasarkan hasil evaluasi, model *CNN* menunjukkan kemampuan yang baik dalam mengklasifikasikan jenis jambu biji dengan tingkat akurasi yang tinggi.

```
Epoch 1/15
13/13 ————— 23s 2s/step - accuracy: 0.7484 - loss: 0.5112 - val_accuracy: 0.9838 - val_loss: 0.
Epoch 2/15
13/13 ————— 16s 1s/step - accuracy: 0.9231 - loss: 0.1680 - val_accuracy: 0.6538 - val_loss: 1.
Epoch 3/15
13/13 ————— 16s 1s/step - accuracy: 0.9279 - loss: 0.1962 - val_accuracy: 0.9615 - val_loss: 0.
Epoch 4/15
13/13 ————— 21s 1s/step - accuracy: 0.9888 - loss: 0.0661 - val_accuracy: 0.9615 - val_loss: 0.
Epoch 5/15
13/13 ————— 13s 1s/step - accuracy: 0.9984 - loss: 0.0251 - val_accuracy: 1.0000 - val_loss: 0.
Epoch 6/15
13/13 ————— 14s 1s/step - accuracy: 0.9952 - loss: 0.0082 - val_accuracy: 1.0000 - val_loss: 0.
Epoch 7/15
13/13 ————— 13s 1s/step - accuracy: 1.0000 - loss: 0.0031 - val_accuracy: 1.0000 - val_loss: 5.
Epoch 8/15
13/13 ————— 13s 1s/step - accuracy: 1.0000 - loss: 8.3345e-04 - val_accuracy: 1.0000 - val_loss
Epoch 9/15
13/13 ————— 14s 1s/step - accuracy: 1.0000 - loss: 8.0175e-04 - val_accuracy: 1.0000 - val_loss
Epoch 10/15
13/13 ————— 14s 1s/step - accuracy: 0.9952 - loss: 0.0064 - val_accuracy: 1.0000 - val_loss: 0.
Epoch 11/15
13/13 ————— 13s 1s/step - accuracy: 0.9984 - loss: 0.0218 - val_accuracy: 1.0000 - val_loss: 0.
```

Gambar 5. Hasil dari proses Training

8. Proses Testing Menggunakan Data

Tahap akhir adalah melakukan pengujian menggunakan data citra acak yang tidak termasuk dalam data latih maupun data validasi. Citra uji diproses oleh sistem dan hasil klasifikasi ditampilkan secara langsung. Hasil pengujian ini menunjukkan bahwa sistem mampu mengidentifikasi jenis jambu biji sesuai dengan

karakteristik citra yang diberikan. Proses ini membuktikan bahwa model *CNN* dapat digunakan untuk identifikasi jenis jambu biji secara *real-time*.

4. SIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan, dapat disimpulkan bahwa metode *Convolutional Neural Network (CNN)* mampu mengidentifikasi jenis jambu biji secara *real-time* dengan tingkat kinerja yang baik. Hasil pengujian menunjukkan bahwa model *CNN* memperoleh tingkat akurasi tertinggi sekitar 95%, dengan penurunan performa sebesar 2–3% pada beberapa skenario pengujian akibat perbedaan kondisi pencahayaan dan kualitas citra. Nilai presisi rata-rata yang dihasilkan berada pada kisaran 94%, sedangkan nilai recall mencapai sekitar 93%, yang menunjukkan bahwa sistem mampu mengenali sebagian besar data uji dengan benar. Selain itu, nilai F1-score berada pada rentang 93–94%, menandakan keseimbangan yang baik antara presisi dan *recall*. Sistem yang dikembangkan memiliki keunggulan dalam hal kecepatan identifikasi dan kemudahan penggunaan karena mampu menampilkan hasil klasifikasi secara langsung melalui kamera. Namun, keterbatasan penelitian ini terletak pada jumlah dataset yang masih terbatas serta variasi kondisi lingkungan yang belum sepenuhnya terwakili. Dengan demikian, penerapan *CNN* pada identifikasi jenis jambu biji secara *real-time* dinilai efektif dan berpotensi untuk dikembangkan lebih lanjut pada penelitian selanjutnya.

5. SARAN

Penelitian selanjutnya disarankan untuk memperluas jumlah dan variasi *dataset* citra jambu biji agar sistem mampu mengenali lebih banyak jenis varietas maupun kondisi buah dengan tingkat akurasi yang lebih tinggi. Selain itu, perlu dilakukan perbandingan dengan metode lain atau pengembangan arsitektur *deep learning* yang lebih kompleks guna memperoleh performa model yang lebih optimal. Pengembangan sistem juga dapat diarahkan pada peningkatan kualitas antarmuka serta integrasi dengan perangkat mobile agar aplikasi dapat digunakan secara lebih luas dan praktis oleh petani maupun pelaku usaha di bidang pertanian, sehingga sistem identifikasi yang dihasilkan dapat memberikan manfaat nyata dalam mendukung proses budidaya dan pengolahan jambu biji.

DAFTAR PUSTAKA

- [1] D. Ayub and D. Fitrilinda, “Budidaya Jambu Biji dalam Meningkatkan Perekonomian Masyarakat (Studi Literature),” vol. 7, pp. 14304–14309, 2024.
- [2] R. A. Nugraha, E. W. Hidayat, N. I. Kurniati, and R. N. Shofa, “Klasifikasi Jenis Buah Jambu Biji Menggunakan Algoritma Principal Component Analysis dan K-Nearest Neighbor,” vol. 7, no. 1, pp. 1–7, 2023.
- [3] C. Engineering, “(Journal of Computer Engineering, System and Science),” vol. 10, no. 2, pp. 517–529, 2025.
- [4] V. No, N. B. Pamungkas, and A. Suhendar, “Edumatic : Jurnal Pendidikan Informatika Penerapan Metode Convolutional Neural Network pada Sistem Klasifikasi Penyakit Tanaman Apel berdasarkan Citra Daun,” vol. 8, no. 2, pp. 675–684, 2024, doi: 10.29408/edumatic.v8i2.27958.
- [5] J. Vicky *et al.*, “Implementasi Pendeteksi Penyakit pada Daun Alpukat Menggunakan Metode CNN,” vol. 2, pp. 155–162, 2023.
- [6] N. Rinanto, R. A. Putri, M. Ardiana, A. Khumaidi, and E. Nafiatu, “Sistem Deteksi dan Pengenalan Wajah Real-Time dengan Haar Cascade dan CNN Untuk Otomatisasi Kehadiran Mahasiswa Automated Student Attendance System Based on Real-Time Face Detection and Recognition Using Haar Cascade and CNN,” vol. 7, no. 2, pp. 203–210, 2025.
- [7] H. K. Zulfiani and A. Abdullah, “IDENTIFIKASI DAUN HERBAL UNTUK TANAMAN OBAT MENGGUNAKAN METODE CONVOLUTIONAL NEURAL NETWORK (CNN),” vol. 9, no. 6, pp. 10429–10435, 2025.
- [8] M. Rizal *et al.*, “DETEKSI AWAL PENYAKIT TANAMAN BERBASIS CITRA DAUN: STUDI PERBANDINGAN ALGORITME SVM DAN CNN,” vol. 1, no. 1, pp. 61–76, 2025.
- [9] F. R. Valerian *et al.*, “Klasifikasi tingkat obesitas menggunakan metode gbm dan confusion matrix,” vol. 9, no. 2, pp. 2242–2249, 2025.
- [10] M. M. Ibrahim, R. Rahmadewi, L. Nurpulaela, F. Teknik, and U. S. Karawang, “PENDETEKSIAN NOMINAL UANG PADA GAMBAR MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORK : INTEGRASI METODE PRA- PEMROSESAN CITRA DAN KLASIFIKASI BERBASIS CNN,” vol. 7, no. 2, pp. 1395–1400, 2023.