



Pengenalan Biji Kopi Berdasarkan Green Bean Menggunakan Convolutional Neural Network (CNN) Berbasis Android

Anisul Fuad Aziz, Nahar Mardiyantoro, Nur Hasanah

Teknik Informatika, Fakultas Teknik dan Ilmu Komputer,

Universitas Sains Al-Qur'an

anisulfuadaziz@gmail.com, nahar@unsiq.ac.id, nurh.unsiq@gmail.com

Diterima : 3 Januari 2026 ; Disetujui : 27 Januari 2026 ; Dipublikasikan : 31 Januari 2026

ABSTRAK

Kabupaten Wonosobo merupakan salah satu daerah penghasil kopi di Jawa Tengah yang memiliki potensi besar, khususnya pada komoditas kopi dalam bentuk green bean. Namun, perbedaan karakteristik visual antar jenis biji kopi seperti Arabika, Robusta, Liberika, dan Ekselsa sering kali sulit dikenali secara kasat mata, terutama oleh masyarakat umum. Penelitian ini bertujuan untuk membangun aplikasi berbasis Android yang mampu mengenali jenis biji kopi berdasarkan citra green bean menggunakan metode Convolutional Neural Network (CNN). Dataset yang digunakan berupa citra biji kopi green bean yang diperoleh melalui proses observasi dan pengambilan gambar langsung dengan pengaturan resolusi, ISO, dan tingkat kecerahan tertentu. Proses penelitian meliputi pengolahan data, augmentasi citra, perancangan arsitektur CNN, pelatihan model menggunakan Google Colaboratory, serta konversi model ke format TensorFlow Lite untuk diimplementasikan pada aplikasi Android menggunakan bahasa pemrograman Kotlin. Evaluasi kinerja model dilakukan menggunakan confusion matrix untuk mengukur akurasi, presisi, dan recall. Selain itu, pengujian sistem dilakukan melalui pengujian blackbox dan usability untuk memastikan fungsi aplikasi berjalan sesuai kebutuhan pengguna. Hasil penelitian menunjukkan bahwa metode CNN mampu mengenali jenis biji kopi green bean dengan baik dan aplikasi yang dikembangkan dapat digunakan sebagai alat bantu identifikasi biji kopi secara cepat dan praktis bagi petani, pelaku usaha, maupun konsumen kopi di Wonosobo.

Kata Kunci : Biji Kopi, Green Bean, Convolutional Neural Network, Android.

ABSTRACT

Wonosobo Regency is one of the coffee-producing regions in Central Java with significant potential, particularly in green bean coffee commodities. However, visual differences among coffee bean types such as Arabica, Robusta, Liberica, and Excelsa are often difficult to distinguish by the naked eye, especially for non-experts. This study aims to develop an Android-based application capable of identifying coffee bean types based on green bean images using the Convolutional Neural Network (CNN) method. The dataset used consists of green bean coffee images obtained through direct observation and image acquisition with specific resolution, ISO, and brightness settings. The research process includes data preprocessing, image augmentation, CNN architecture design, model training using Google Colaboratory, and model conversion into TensorFlow Lite format for implementation in an Android application developed using the Kotlin programming language. Model performance evaluation is conducted using a confusion matrix to measure accuracy, precision, and recall. In addition, system testing is carried out through blackbox testing and usability testing to ensure that the application functions according to user requirements. The results indicate that the CNN method is able to effectively recognize green bean coffee types, and the developed application can serve as a practical and efficient identification tool for farmers, business actors, and coffee consumers in Wonosobo.

Keywords : Coffee Bean, Green Bean, Convolutional Neural Network, Android

1. PENDAHULUAN

Kabupaten Wonosobo merupakan salah satu daerah di Provinsi Jawa Tengah yang terletak di kawasan dataran tinggi Dieng dengan ketinggian sekitar 750 hingga 2.250 meter di atas permukaan laut. Kondisi geografis Wonosobo yang dikelilingi pegunungan seperti Gunung Sumbing, Gunung Sindoro, dan Gunung Prau menjadikan wilayah ini memiliki tanah yang subur serta iklim yang sejuk. Faktor tersebut sangat mendukung perkembangan sektor pertanian, terutama tanaman perkebunan seperti kopi [1].

Tanaman kopi menjadi salah satu komoditas utama yang ada di Wonosobo karena memiliki nilai ekonomis yang tinggi [2]. Kopi Wonosobo kini tidak hanya dikenal di pasar lokal tetapi juga mulai dilirik oleh pasar nasional dan internasional dengan kopi yang sudah terkenal yaitu Kopi Bowongso.

Green bean adalah biji kopi dalam bentuk mentah yang sudah dikeringkan dan siap disangrai, memiliki berbagai jenis yang dibedakan berdasarkan varietas tanaman serta daerah tumbuhnya atau yang biasa disebut dengan [3]. Beberapa jenis green bean yang banyak dikenal antara lain Arabika, Robusta, Ekselsa dan Liberika. Setiap jenis kopi tersebut memiliki ciri khas tersendiri baik dari bentuk, ukuran, warna, maupun karakter rasa [4]. Misalnya, kopi Arabika dikenal memiliki rasa asam yang lembut dengan aroma buah-buahan, sedangkan kopi Robusta cenderung memiliki rasa yang lebih pahit dan kadar kafein yang lebih tinggi [5]. Perbedaan ini menjadikan proses pengenalan jenis green bean menjadi penting dalam menentukan kualitas dan cita rasa kopi yang akan dihasilkan.

Namun demikian, dalam proses pengolahan dan perdagangan biji kopi, terutama dalam bentuk green bean, sering kali para peminat atau pembeli mengalami kesulitan dalam mengenali jenis kopi secara visual [3]. Hal ini disebabkan karena bentuk, ukuran, warna, dan tekstur green bean antar jenis kopi sering kali tampak hampir sama sehingga sulit dibedakan secara kasat mata, terutama bagi orang awam [3]. Akibatnya, proses identifikasi jenis kopi sering kali bergantung pada pengalaman atau uji rasa. Kondisi ini tentu menyulitkan bagi konsumen yang ingin memastikan jenis kopi secara cepat dan akurat.

Untuk mengatasi permasalahan tersebut, diperlukan solusi yang dapat membantu dalam mengenali jenis biji kopi Wonosobo berdasarkan citra green bean dengan pendekatan menggunakan metode Convolutional Neural Network (CNN). Berdasarkan penelitian berjudul “Perancangan Klasifikasi Tanaman Herbal Menggunakan Transfer Learning pada Algoritma Convolutional Neural Network (CNN)” menghasilkan sebuah aplikasi yang dapat mengenali tanaman herbal berdasarkan citra daun menggunakan CNN [6]. Selain itu, pada penelitian berjudul “Klasifikasi Jenis Ikan Guppy Menggunakan Convolutional Neural Network (CNN)” membuktikan bahwa CNN dapat melakukan klasifikasi berdasarkan pola warna dari ikan [7]. Adapun penelitian dari Deni Hidayat yang berjudul “Klasifikasi Jenis Mangga Berdasarkan Bentuk dan Tekstur Daun Menggunakan Metode Convolutional Neural Network (CNN)” membuktikan bahwa CNN dapat mengklasifikasi jenis mangga melalui bentuk daunnya [8]. Hal ini membuktikan bahwa pemilihan metode Convolutional Neural Network untuk mengenali jenis biji kopi Wonosobo berdasarkan green bean.

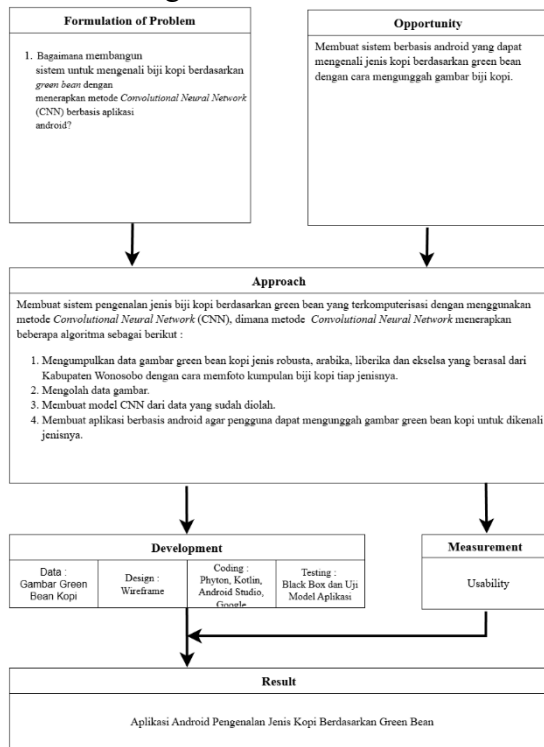
Berdasarkan latar belakang tersebut, maka dibuatlah penelitian dengan judul “Pengenalan Biji Kopi Berdasarkan Green Bean Menggunakan Convolutional Neural Network (CNN) Berbasis Android Studi Kasus di Wonosobo” dengan mengembangkan sebuah aplikasi android yang mampu mengidentifikasi jenis biji kopi Wonosobo berdasarkan green bean menggunakan metode Convolutional Neural Network melalui citra atau foto dari kamera ataupun galeri.

2. METODE

Penelitian ini tergolong dalam jenis penelitian eksperimental. Penelitian eksperimental merupakan metode penelitian yang bertujuan untuk menguji suatu hipotesis dengan cara memanipulasi variabel independen serta mengamati dampaknya terhadap variabel dependen dalam kondisi yang terkontrol. Dalam konteks penelitian ini, tujuan utamanya adalah untuk mengetahui kemampuan algoritma Convolutional Neural Network (CNN) dalam mengenali jenis tembakau rajangan Temanggung berdasarkan citra atau foto. Objek penelitian ini mencakup proses pemilihan

arsitektur model, pelatihan, serta evaluasi kinerja model CNN.

Agar mencapai tujuan dari penelitian, peneliti merencanakan kerangka konseptual penelitian sebagai berikut.



Gambar 1. Kerangka Konseptual

2.1. Pengumpulan Data

Peneliti mengumpulkan data terkait biji kopi berupa gambar biji kopi dengan jenis Arabika, Robusta, Liberika, dan Ekselsa. Metode yang digunakan dalam pengumpulan data yang digunakan yaitu observasi dan wawancara.

Dimana hasil observasi dilakukan dengan mendatangi sejumlah petani dan pedagang biji kopi di Wonosobo memperoleh data berupa foto dari setiap jenis biji kopi, yaitu Arabika, Robusta, Ekselsa, dan Liberika. Dari hasil pengamatan tersebut, berhasil dikumpulkan sebanyak 1000 gambar, yang terdiri atas 250 foto jenis Arabika, 250 foto jenis Robusta, 250 foto jenis Ekselsa, dan 250 foto jenis Liberika yang diambil menggunakan kamera handphone Iphone 16 kamera utama 48MP dengan ketentuan iso yang digunakan saat pengambilan gambar yaitu 64 – 125 dan tingkat kecerahan gambar 4 – 7 serta ukuran gambar yang dihasilkan yaitu 3024 × 4032, dimana pengambilan gambar yaitu kumpulan biji kopi *green bean* berlatar belakang warna putih.

Hasil wawancara dengan Wawancara dilakukan untuk memperoleh informasi

mengenai karakteristik dari setiap jenis biji kopi. Informasi yang didapatkan bisa berupa rasa, aroma, bentuk, dan harga dari setiap jenis biji kopi. Sasaran wawancara ini adalah para petani atau penjual biji kopi. Dimana dari data hasil wawancara menghasilkan data sebagai berikut.

Tabel 1. Data Hasil Wawancara

No	Jenis Biji Kopi	Deskripsi
1.	Arabika	Arabika memiliki biji berbentuk lonjong dengan lekukan tengah menyerupai huruf S. Jenis ini dikenal memiliki rasa halus dan kompleks, dengan Tingkat keasaman tinggi, serta aroma fruity. Arabika umumnya dibudidayakan pada ketinggian 900-2000 mdpl sehingga membutuhkan suhu yang lebih sejuk.
2.	Robusta	Robusta memiliki bentuk biji yang lebih bulat dengan lekukan lurus di bagian Tengah. Rasanya cenderung lebih pahit, berbody tebal, dan memiliki keasaman rendah. Kandungan kafein yang tinggi membuat cita rasa Robusta lebih kuat dan kopi ini cocok ditanam pada ketinggian 400-900 mdpl, serta tahan hama.
3.	Liberika	Liberika memiliki biji berukuran besar, bentuknya tidak simetris, dan cenderung menggelembung. Rasa Liberika dikenal unik dengan karakter woody, smoky, dan nutty, serta body yang cukup tebal. Kopi ini biasanya tumbuh pada ketinggian 200-700 mdpl dan mampu beradaptasi dengan lingkungan lembab.
4.	Ekselsa	Ekselsa masih dikategorikan sebagai varietas dari Liberika, memiliki biji besar namun lebih ramping dan lonjong. Rasanya lebih fruity dengan aroma kompleks, terkadang memiliki after taste winey atau fermentatif. Ekselsa cocok tumbuh pada ketinggian 200-800 mdpl dan dikenal tahan terhadap kondisi lahan yang keras.

2.2. Metode Analisis Data

Penelitian ini menerapkan metode *Convolutional Neural Network* (CNN) untuk melakukan klasifikasi jenis biji kopi berdasarkan *green bean* menggunakan citra melalui aplikasi berbasis Android. CNN merupakan salah satu cabang dari Deep Learning yang sangat efektif dalam memproses dan menganalisis data visual seperti gambar. Data citra yang diperoleh dikumpulkan dan dikelompokkan sesuai jenisnya ke dalam beberapa folder.

Proses pengolahan data dilakukan menggunakan Google Collaboratory sebagai platform penulisan dan eksekusi kode. Tahapan awal meliputi pre-processing dan augmentasi data. Pada tahap pre-processing, dilakukan *rescaling* dengan mengubah rentang nilai piksel dari 0–255 menjadi 0–1 untuk mempercepat komputasi, serta *resize* agar seluruh gambar memiliki ukuran piksel yang seragam. Selanjutnya, dilakukan augmentasi data melalui teknik *zoom range*, *rotation*, *vertical flip*, dan *horizontal flip* guna memperbanyak variasi citra yang digunakan dalam pelatihan.

Setelah proses tersebut, dataset dibagi menjadi 80% data pelatihan dan 20% data validasi. Tahap berikutnya adalah pelatihan model menggunakan bahasa pemrograman Python dengan memanfaatkan data pelatihan. Model yang dihasilkan kemudian diuji menggunakan data validasi untuk mengevaluasi performanya dalam mengenali citra. Hasil pengujian dianalisis menggunakan *confusion matrix* guna memperoleh nilai akurasi, presisi, dan *recall* dari model yang telah dibangun.

2.3. Pengolahan Data

Pengolahan data hasil observasi adalah gambar biji kopi sebanyak 1000 gambar dengan pembagian yaitu 250 gambar untuk setiap jenis biji kopinya. Dimana dari data tersebut akan dikumpulkan dalam folder berdasarkan jenis biji kopinya. Kemudian folder dataset akan diubah menjadi folder berbentuk zip dan diunggah ke Google Drive.

Data dari hasil wawancara merupakan informasi terkait ciri-ciri dari jenis biji kopi. Dimana data wawancara akan dijadikan sebagai informasi jenis biji. Informasi ini akan menjadi deskripsi dari biji kopi pada aplikasi android yang dibuat.

2.4. Pre-processing

Tahap *pre-processing* merupakan langkah awal dalam pengolahan citra pada dataset yang bertujuan untuk menghasilkan data yang lebih optimal dan efisien sebelum digunakan dalam proses pelatihan model. Pada tahap ini, setiap citra diberikan beberapa perlakuan, yaitu *resize* dan *rescaling*. Proses *resize* dilakukan untuk menyesuaikan ukuran serta resolusi citra agar sesuai dengan kebutuhan sistem, sedangkan *rescaling* bertujuan untuk menormalkan nilai piksel ke dalam rentang tertentu sehingga dapat mempermudah dan mempercepat proses komputasi.

2.5. Augmentasi Dataset

Augmentasi dataset merupakan tahapan pengolahan citra yang bertujuan untuk meningkatkan jumlah data dengan melakukan berbagai perubahan pada bentuk maupun posisi gambar. Penerapan teknik augmentasi ini dilakukan agar model yang dikembangkan memiliki kemampuan generalisasi yang lebih baik serta mampu mengurangi terjadinya *overfitting*. *Overfitting* adalah kondisi ketika model terlalu menyesuaikan pola pada data pelatihan sehingga performanya menurun saat dihadapkan pada data yang belum pernah dilihat sebelumnya. Selanjutnya, dataset hasil augmentasi tersebut dibagi menjadi data pelatihan (*training*) dan data validasi (*validation*).

2.6. Pelatihan Model

Pelatihan model merupakan tahapan pembelajaran mesin yang memanfaatkan data pelatihan untuk membangun sebuah model klasifikasi. Pada proses ini, model terlebih dahulu dikompilasi dengan menggunakan *optimizer* Adam, metrik evaluasi berupa *accuracy*, serta fungsi *loss categorical crossentropy* yang dipilih karena dataset yang digunakan terdiri dari empat kelas.

2.7. Evaluasi Kinerja Model

Dalam pengembangan model machine learning, model yang telah jadi perlu dilakukan sebuah evaluasi kinerja model terutama dalam konteks klasifikasi multi kelas dan klasifikasi biner. Alat yang dapat digunakan dalam evaluasi kinerja model adalah *confusion matrix*. *Confusion matrix* akan menyajikan jumlah prediksi benar dan salah yang dilakukan oleh model pada data pengujian.

Secara umum, confusion matrix memiliki 4 sel, yang mewakili 4 kemungkinan hasil dari proses klasifikasi yaitu :

1) True Positive (TP)

True positive menunjukkan jumlah sampel yang diklasifikasi dengan benar sebagai positif oleh model.

2) False Positive (FP)

False positive menunjukkan jumlah sampel yang salah diklasifikasikan sebagai positif oleh model. Terkadang disebut Type I Error.

3) True Negative (TN)

True negative menunjukkan jumlah sampel yang diklasifikasi dengan benar sebagai negative oleh model.

4) False Negative (FN)

False negative menunjukkan jumlah sampel yang salah diklasifikasikan sebagai negatif oleh model. Terkadang disebut Type II Error.

Menurut 4 sel kemungkinan yang ada, dapat direpresentasikan dalam tabel berikut.

Tabel 2. Confusion Matrix

Confusion Matrix	Predicted Values		
		Positive	Negative
Actual Values	Positive	TP	FP
	Negative	FN	TN

Dari confusion matrix, dapat menghitung berbagai matrix evaluasi seperti akurasi, presisi, dan *recall*.

1. Akurasi

Akurasi mengukur seberapa sering model benar dalam melakukan prediksi, dinyatakan sebagai presentase dari total prediksi yang benar.

2. Presisi

Presisi mengukur seberapa sering model memberikan prediksi positif yang benar, dinyatakan sebagai presentase dari total prediksi positif yang benar.

3. Recall

Recall mengukur seberapa sering model dapat mendeteksi kelas positif dengan benar, dinyatakan sebagai presentase dari total sampel kelas positif yang diidentifikasi dengan benari.

2.8. Pengembangan Aplikasi Android

Dalam proses pengembangan aplikasi Android digunakan IDE (*Integrated Development Environment*) Android Studio dengan bahasa pemrograman Kotlin. Untuk menghubungkan aplikasi Android dengan model yang diintegrasikan ke dalam aplikasi,

digunakan bantuan pustaka Keras dan TensorFlow sebagai pendukung proses integrasi model.

3. HASIL DAN PEMBAHASAN

3.1. Pembuatan Model Arsitektur CNN

Arsitektur model CNN dimulai dari konvolusi sampai klasifikasi. Dimana arsitektur yang dibuat yaitu terjadi konvolusi sebanyak 4 kali, menggunakan padding same, aktivasi menggunakan ReLU, *pooling layer* menggunakan *max-pooling*, *dropout* sebanyak 50%, *flattening*, dan klasifikasi. Kode untuk arsitektur dari model CNN dapat dilihat pada gambar 14 dibawah ini.

```

model = tf.keras.models.Sequential([
    Conv2D(16, (3,3), activation='relu', padding='same', input_shape=(TARGET_SIZE, TARGET_SIZE, 3)),
    MaxPooling2D(2,2),
    Conv2D(32, (3,3), activation='relu', padding='same'),
    MaxPooling2D(2,2),
    Conv2D(64, (3,3), activation='relu', padding='same'),
    MaxPooling2D(2,2),
    Conv2D(128, (3,3), activation='relu', padding='same'),
    MaxPooling2D(2,2),
    Flatten(),
    Dropout(0.5),
    Dense(128, activation='relu'),
    Dense(len(class_names), activation='softmax')
])
model.summary()
    
```

Gambar 2. Kode Pembuatan Arsitektur CNN

Kode tersebut menghasilkan arsitektur model Convolutional Neural Network (CNN) yang dibangun secara sekuensial menggunakan *framework* Keras dan dirancang untuk keperluan klasifikasi citra. Struktur model terbagi ke dalam dua komponen utama, yaitu tahap ekstraksi fitur dan tahap klasifikasi. Pada bagian ekstraksi fitur, model terdiri dari empat blok konvolusi yang disusun secara berurutan, dengan jumlah filter yang meningkat secara bertahap mulai dari 16 hingga 128 pada blok terakhir. Setiap blok mengombinasikan lapisan Conv2D dengan ukuran kernel 3×3 dan fungsi aktivasi ReLU, yang kemudian diikuti oleh lapisan MaxPooling2D dengan ukuran *pool* 2×2 . Pola ini memungkinkan model untuk menangkap fitur-fitur citra secara bertingkat, dari fitur sederhana hingga lebih kompleks, sekaligus mengurangi dimensi spasial dari data masukan.

Pada bagian klasifikasi, keluaran dari tahap ekstraksi fitur terlebih dahulu diratakan menggunakan lapisan Flatten untuk mengubah representasi tiga dimensi menjadi vektor satu dimensi. Selanjutnya, lapisan *dropout* dengan nilai 0.5 diterapkan untuk mengurangi risiko terjadinya overfitting. Vektor hasil proses ini kemudian diteruskan ke lapisan *dense* dengan

128 neuron dan fungsi aktivasi ReLU, sebelum akhirnya diproses oleh lapisan keluaran *dense* yang menggunakan fungsi aktivasi *softmax*. Lapisan keluaran ini memiliki jumlah neuron yang disesuaikan dengan jumlah kelas, sehingga mampu menghasilkan probabilitas untuk setiap kelas sebagai dasar penentuan prediksi akhir. Untuk melihat hasil arsitektuk CNN dapat dilihat pada gambar 15 dibawah ini.

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 299, 299, 16)	448
max_pooling2d (MaxPooling2D)	(None, 149, 149, 16)	0
conv2d_1 (Conv2D)	(None, 149, 149, 32)	4,648
max_pooling2d_1 (MaxPooling2D)	(None, 74, 74, 32)	0
conv2d_2 (Conv2D)	(None, 74, 74, 64)	18,496
max_pooling2d_2 (MaxPooling2D)	(None, 37, 37, 64)	0
conv2d_3 (Conv2D)	(None, 37, 37, 128)	73,856
max_pooling2d_3 (MaxPooling2D)	(None, 18, 18, 128)	0
flatten (Flatten)	(None, 41472)	0
dropout (Dropout)	(None, 41472)	0
dense (Dense)	(None, 128)	5,308,544
dense_1 (Dense)	(None, 4)	516

Total params: 5,406,500 (20.62 MB)
 Trainable params: 5,406,500 (20.62 MB)
 Non-trainable params: 0 (0.00 B)

Gambar 3. Arsitektur CNN

3.2. Pelatihan Dataset

Sebelum dilakukan proses pelatihan, maka buat sebuah aturan yang bisa memberhentikan proses pelatihan. Dimana proses pelatihan akan berhenti ketika akurasi dan validasi akurasi sudah mencapai 98%. Untuk membuat aturan tersebut dapat dilakukan dengan menjalankan kode pada gambar 16 dibawah ini.

```
report_datetime

class MyCallback(tf.keras.callbacks.Callback):
    def on_epoch_end(self, epoch, logs={}):
        if(logs.get('accuracy') >= 0.98 and logs.get('val_accuracy') >= 0.98):
            print("\nTraining sudah mencapai akurasi diatas 98%")
            self.model.stop_training = True
callbacks = MyCallback()

log_dir = "logs/fit/" + datetime.datetime.now().strftime("%Y%m%d-%H%M%S")
fm = tf.summary.create_file_writer("./logs/")
tf.summary.create_file_writer("./logs/")
file_writer_cm = tf.summary.create_file_writer(log_dir + '/cm')
tensorboard_callback = tf.keras.callbacks.TensorBoard(log_dir=log_dir, histogram_freq=1)
```

Gambar 4. Kode Callback Pelatihan Dataset

Setelah arsitektur CNN sudah dibuat maka proses pelatihan dataset dapat dijalankan. Dimana kode untuk memulai proses pelatihan dapat dilihat pada gambar 17 dibawah ini.

```
model.compile(
    loss='categorical_crossentropy',
    optimizer=tf.keras.optimizers.Adam(),
    metrics=['accuracy']
)
```

Gambar 5. Kode Mulai Pelatihan Dataset

Pada tahap pelatihan, model terlebih dahulu dikompilasi menggunakan fungsi `model.compile` untuk menetapkan parameter-parameter utama yang dibutuhkan dalam proses training. Fungsi kerugian yang digunakan adalah Categorical Cross-Entropy, yang sesuai untuk permasalahan klasifikasi multi-kelas dengan label dalam bentuk *one-hot encoding*, dan berperan dalam mengukur selisih antara probabilitas hasil prediksi dengan nilai target yang sebenarnya. Proses optimasi dilakukan menggunakan algoritma Adam (*Adaptive Moment Estimation*), yang dikenal mampu menyesuaikan laju pembelajaran secara adaptif pada setiap parameter sehingga proses konvergensi menjadi lebih cepat dan stabil. Sementara itu, metrik akurasi (*accuracy*) digunakan sebagai indikator kinerja model, yang dipantau selama tahap pelatihan maupun pengujian untuk mengetahui tingkat ketepatan prediksi yang dihasilkan. Setelah proses training selesai maka akan menghasilkan keterangan dapat dilihat pada gambar 18 dibawah ini.



Gambar 6. Hasil Pelatihan Dataset

3.3. Pengujian Akurasi dan Loss Model

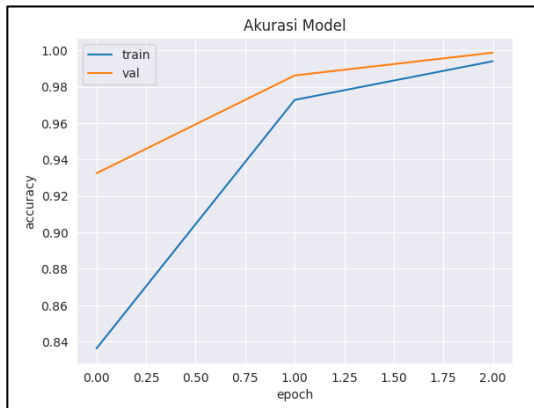
Setelah proses pelatihan selesai maka dapat menghasilkan sebuah grafik akurasi model yang menunjukkan akurasi model selama proses pelatihan dan grafik loss model yang menunjukkan loss model selama proses pelatihan. Kode untuk menghasilkan grafik akurasi dan loss model dapat dilihat pada gambar 19 dibawah ini.

```
plt.plot(history.history["accuracy"])
plt.plot(history.history["val_accuracy"])
plt.title("Akurasi Model")
plt.ylabel("accuracy")
plt.xlabel("epoch")
plt.legend(["train", "val"], loc="upper left")
plt.show()

plt.plot(history.history["loss"])
plt.plot(history.history["val_loss"])
plt.title("Loss Model")
plt.ylabel("loss")
plt.xlabel("epoch")
plt.legend(["train", "val"], loc="upper left")
plt.show()
```

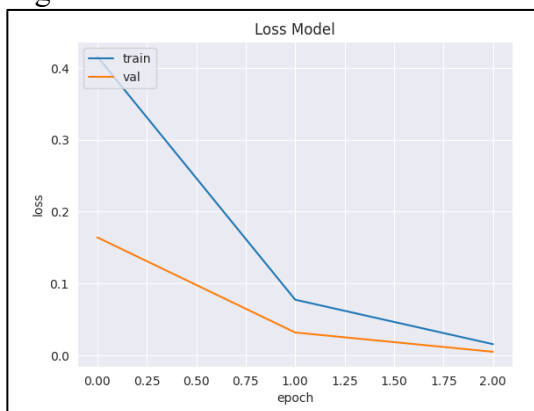
Gambar 7. Kode Pembuatan Grafik Akurasi dan Loss Model

Dari pelatihan dataset sampai menjadi sebuah model, menghasilkan sebuah grafik akurasi model dapat dilihat pada gambar 20 dibawah ini.



Gambar 8. Grafik Akurasi Model

Berdasarkan grafik akurasi model, visualisasi yang membandingkan kurva akurasi data pelatihan (train) dan validasi (val) terhadap jumlah *epoch* menunjukkan bahwa proses pelatihan mengalami konvergensi yang sangat cepat. Akurasi pada data pelatihan, yang ditunjukkan oleh garis biru, meningkat secara konsisten dari nilai awal sekitar 0,64 hingga mencapai hampir sempurna, yaitu 0,99 pada epoch kedua. Sementara itu, akurasi data validasi yang direpresentasikan oleh garis oranye mengalami peningkatan yang lebih signifikan, dengan nilai akurasi awal sebesar 0,93 pada epoch pertama dan terus meningkat hingga mencapai akurasi validasi sebesar 0,98. Selain grafik akurasi model, model akan menghasilkan grafik loss model dapat dilihat pada gambar 21 dibawah ini.



Gambar 9. Grafik Loss Model

Berdasarkan grafik loss model, terlihat bahwa proses pelatihan menunjukkan penurunan nilai loss yang signifikan. Pada data pelatihan, nilai loss menurun dari 0,87 pada epoch pertama menjadi 0,01 pada epoch kedua. Sementara itu, pada data validasi juga terjadi

penurunan loss, dari nilai 0,164 pada epoch pertama menjadi 0,004 pada epoch kedua. Pola ini menunjukkan bahwa model mampu meminimalkan kesalahan secara efektif baik pada data pelatihan maupun data validasi.

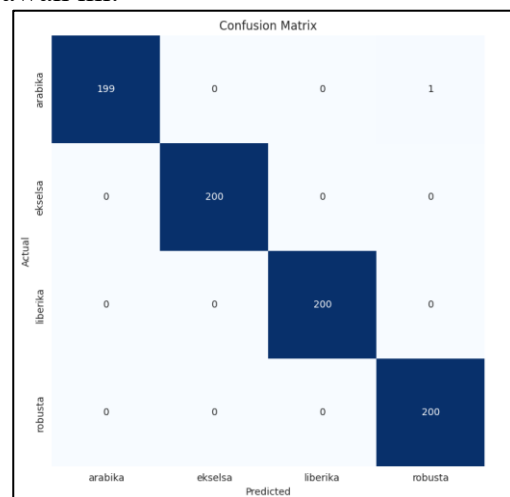
3.4. Confusion Matrix

Setelah model sudah dibuat maka perlu dilakukan proses evaluasi model untuk mengetahui kinerja dari model tersebut. Proses evaluasi ini dilakukan dengan menggunakan dataset validasi untuk menguji seberapa baik model dalam mengenali gambar. Dimana proses evaluasi model ini akan direpresentasikan menjadi sebuah *confusion matrix*. Untuk kode pembuatan confusion matrix dapat dilihat pada gambar 22 dibawah ini.

```
def on_epoch_end(generator, model, class_names):
    # prediction generator (runs all model steps)
    preds = model.predict(generator, verbose=1)
    y_pred = np.argmax(preds, axis=1)
    y_true = generator.labels # works because shuffle=False for validation generator
    target_names = class_names
    print("Classification Report:\n", classification_report(y_true, y_pred, target_names=target_names))
    cm = confusion_matrix(y_true, y_pred)
    plt.figure(figsize=(10,10))
    sns.heatmap(cm, annot=True, fmt='g', cmap='Blues', cbar=False,
                xticklabels=target_names, yticklabels=target_names)
    plt.xlabel("Predicted")
    plt.ylabel("Actual")
    plt.title("Confusion Matrix")
    plt.show()
on_epoch_end(validation_generator, model, class_names)
```

Gambar 10. Kode Pembuatan Confusion Matrix

Dimana setelah menjalankan kode tersebut akan menampilkan hasil *confusion matrix* dari model. Dari pengujian model yang sudah dibuat terhadap dataset validasi, model menghasilkan *confusion matrix* dapat dilihat pada gambar 23 dibawah ini.

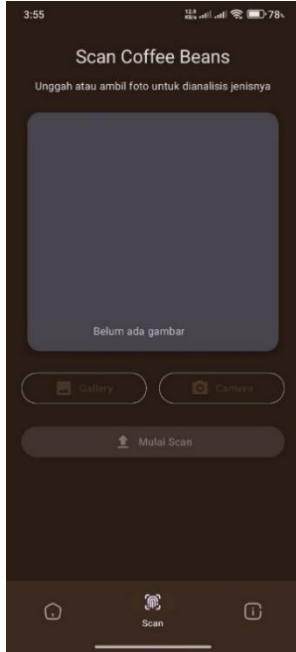


Gambar 11. Confusion Matrix

Berdasarkan hasil *confusion matrix*, model menunjukkan saat diuji dengan data validasi berjumlah 800 gambar dengan tiap-tiap jenis berjumlah 200 gambar berhasil menunjukkan kesesuaian klasifikasi berjumlah 799 gambar dengan klasifikasi salah 1 gambar yang seharusnya arabika tetapi masuk ke robusta.

3.5. Pengembangan Aplikasi Android

Ketika pengguna menekan icon *scan* di home, maka akan tampil halaman *scan* dengan tombol mulai *scan* dalam keadaan tidak aktif. Hal ini dikarenakan pengguna belum memilih gambar dari galeri ataupun kamera. Tampilan halaman *scan* dapat dilihat pada gambar 28 dibawah ini.



Gambar 12. Tampilan Halaman Scan

Setelah pengguna memilih gambar dari galeri maupun kamera, maka gambar yang dipilih akan tampil di halaman *scan* dan tombol mulai *scan* sudah dapat ditekan.

Setelah proses scan selesai akan menampilkan hasil scan berupa kondisi, nilai konfidensi, dan deskripsi. Tampilan halaman hasil deteksi dapat dilihat pada gambar 29 dibawah ini.



Gambar 13. Tampilan Halaman Hasil Scan

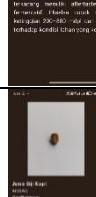
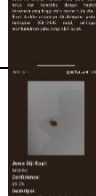
3.6. Pengujian Model Pada Aplikasi

Pengujian model terhadap aplikasi dilakukan untuk mengetahui seberapa sering model dapat mengklasifikasi gambar dengan benar. Dimana pengujian menggunakan 20 gambar dengan masing-masing kelas berjumlah 5 gambar. Untuk pengujian *usability* dapat dilihat pada table 3 dibawah ini.

Tabel 3. Pengujian Model Pada Aplikasi

Sumber Gambar	Prediksi	Hasil	Bukti	Keterangan
Galeri	Arabika	Arabika		Sesuai
Galeri	Arabika	Arabika		Sesuai
Galeri	Robusta	Robusta		Sesuai
Galeri	Robusta	Robusta		Sesuai
Galeri	Liberika	Liberika		Sesuai
Galeri	Liberika	Liberika		Sesuai
Galeri	Ekselsa	Ekselsa		Sesuai

Galeri	Ekselsa	Ekselsa		Sesuai
Galeri	Arabika	Arabika		Sesuai
Galeri	Robusta	Robusta		Sesuai
Galeri	Liberika	Arabika		Tidak Sesuai
Galeri	Ekselsa	Arabika		Tidak Sesuai
Kamera	Arabika	Arabika		Sesuai
Kamera	Robusta	Robusta		Sesuai
Kamera	Liberika	Liberika		Sesuai

Kamera	Ekselsa	Ekselsa		Tidak Sesuai
Kamera	Arabika	Arabika		Sesuai
Kamera	Robusta	Robusta		Sesuai
Kamera	Liberika	Arabika		Tidak Sesuai
Kamera	Ekselsa	Arabika		Tidak Sesuai

4. PENUTUP

4.1. Kesimpulan

Berdasarkan hasil penelitian, menghasilkan aplikasi berbasis android yang dapat mengenali 4 jenis biji kopi yaitu Arabika, Robusta, Liberika, dan Ekselsa berdasarkan green bean berasal dari Kabupaten Wonosobo yang menerapkan metode Convolutional Neural Network (CNN) dengan pengujian usability sebanyak 20 gambar uji yang berasal dari kamera dan galeri, aplikasi mampu mengenali gambar biji kopi dengan benar sebanyak 16 kali, sehingga diperoleh tingkat kesesuaian sebesar 80%.

4.2. Saran

Saran Berdasarkan hasil penelitian, saran untuk penelitian selanjutnya adalah aplikasi dapat ditambahkan fitur pengenalan biji kopi berdasarkan tingkat roasting, seperti light roast, medium roast, dan dark roast. Penambahan fitur

ini diharapkan dapat memperluas fungsi aplikasi tidak hanya untuk mengenali jenis biji kopi berdasarkan green bean, tetapi juga untuk mengenali karakteristik biji kopi setelah proses pemanggangan, sehingga aplikasi menjadi lebih komprehensif dan bermanfaat bagi pelaku industri kopi maupun konsumen.

5. DAFTAR PUSTAKA

- [1] Supriyadi, S. (2022). Pemberdayaan Kelompok Tani Hutan Desa Keseneng, Kecamatan Mojotengah, Kabupaten Wonosobo sebagai Desa Penghasil Kopi. SEMAR (Jurnal Ilmu Pengetahuan, Teknologi, Dan Seni Bagi Masyarakat), 11(1), 8–11.
- [2] Ratri, D. K., Wicaksono, I. A., & Utami, D. P. (2022). Strategi Pengembangan Pasar Produk Kopi Arabika Bowongso Berbasis Business Model Canvas (Studi Kasus di Kelompok Tani Bina Sejahtera Desa Bowongso Kecamatan Kalikajar Kabupaten Wonosobo). Surya Agritama: Jurnal Ilmu Pertanian Dan Peternakan, 11(1), 61–77.
- [3] Nugroho, M. A., & Sebatubun, M. M. (2020). Klasifikasi Varietas Kopi Berdasarkan Green Bean Coffee Menggunakan Metode Machine Learning. Journal of Information System Management (JOISM), 1(2), 1–5.
- [4] Sebatubun, M. M., & Pujiarini, E. H. (2018). Pengenalan Varietas Kopi Arabika Berdasarkan Fitur Bentuk. JIKO (Jurnal Informatika Dan Komputer), 3(2), 60–69.
- [5] Herlinawati, L. (2020). Mempelajari pengaruh konsentrasi maltodekstrin dan Polivinil Pirolidon (PVP) terhadap karakteristik sifat fisik tablet effervescent Kopi Robusta (*Coffea robusta* Lindl). Agritekhnologi (Jurnal Agribisnis Dan Teknologi Pangan), 1(01), 1–25.
- [6] Purwanto, P., & Sumardi, S. (2022). Perancangan Klasifikasi Tanaman Herbal Menggunakan Transfer Learning Pada Algoritma Convolutional Neural Network (CNN). Jurnal Ilmiah Infokam, 18(2), 105–118.
- [7] Marshanda, F., Santoso, B., & Armanto, A. (2025). Klasifikasi Jenis Ikan Guppy Menggunakan Convolutional Neural Network (CNN). ESCAF, 967–975.
- [8] Hidayat, D. (2022). Klasifikasi Jenis Mangga Berdasarkan Bentuk Dan Tekstur Daun Menggunakan Metode Convolutional Neural Network (CNN). J. Inf. Technol. Comput. Sci, 5(1), 98–103.