

## SISTEM PENDETEKSI TINGKAT KESEGERAN DAGING AYAM PADA CITRA MENGGUNAKAN METODE CONVOLUTIONAL NEURAL NETWORK (CNN) BERBASIS ANDROID

Rayhan Naturizal✉, Wahyu Fuadi, Lidya Rosnita

Teknik Informatika, Fakultas Teknik, Universitas Malikussaleh, Aceh Utara, Indonesia

Email: [rayhan.200170012@mhs.unimal.ac.id](mailto:rayhan.200170012@mhs.unimal.ac.id)

DOI: [https://doi.org/10.46880/jmika.Vol8No2\\_pp301-312](https://doi.org/10.46880/jmika.Vol8No2_pp301-312)

### ABSTRACT

*This research develops a chicken meat freshness detection system based on image processing, implemented on an Android platform using the Convolutional Neural Network (CNN) method optimized with TensorFlow Lite. The system classifies chicken meat into three categories: fresh, less fresh, and rotten. The CNN model uses 32 filters to enhance feature extraction from the meat images. Testing on 30 samples, with each category tested 10 times, showed an accuracy of 90%, with 27 correct detections and 3 errors in the less fresh category. While the system effectively identifies fresh and rotten categories, there is a challenge in distinguishing the less fresh category due to its ambiguous visual characteristics. One limitation is the lack of a bounding box, causing the application to still provide detection results even when the scanned object is not chicken meat. This application is specifically designed to detect chicken meat pieces, so it is not recommended for use outside this context.*

**Keyword:** CNN, Detection, Freshness, Chicken, Application.

### ABSTRAK

*Penelitian ini mengembangkan sistem deteksi kesegaran daging ayam berbasis citra pada platform Android menggunakan metode Convolutional Neural Network (CNN) yang dioptimalkan dengan TensorFlow Lite. Sistem ini mengklasifikasikan daging ayam ke dalam tiga kategori: segar, kurang segar, dan busuk. Model CNN menggunakan 32 filter untuk meningkatkan ekstraksi fitur dari citra daging. Pengujian pada 30 sampel dengan masing-masing kategori diuji 10 kali menunjukkan akurasi sebesar 90%, dengan 27 deteksi tepat dan 3 kesalahan pada kategori kurang segar. Meskipun sistem efektif dalam mengidentifikasi kategori segar dan busuk, terdapat tantangan dalam membedakan kategori kurang segar karena karakteristik visualnya yang ambigu. Kekurangan sistem ini termasuk tidak adanya bounding box, sehingga aplikasi tetap memberikan hasil deteksi meskipun objek yang dipindai bukan daging ayam. Aplikasi ini dirancang khusus untuk mendeteksi potongan daging ayam, sehingga tidak dianjurkan digunakan di luar konteks tersebut.*

**Kata Kunci:** CNN, Deteksi, Kesegaran, Ayam, Aplikasi.

### PENDAHULUAN

Teknologi terus maju di era modern ini dan memberikan dampak yang sangat signifikan terhadap kemajuan di berbagai bidang, termasuk bidang bahan makanan salah satunya yaitu daging ayam. Saat ini pendeteksi tingkat kesegaran daging ayam masih banyak dilakukan secara manual. Cara ini umumnya berdasarkan pengamatan visual pengecekan aroma dan tekstur daging. Metode ini memiliki keterbatasan dalam beberapa hal. Pertama dalam hal akurasi, sulit untuk mendeteksi perubahan kecil pada kesegaran daging secara manual. Yang kedua objektivitas, penilaian subjektif dapat dipengaruhi oleh faktor eksternal seperti penerangan, kelelahan, atau kekurangan indra penglihatan dan penciuman manusia.

Yang ketiga efisiensi, proses manual membutuhkan waktu dan tenaga lebih.

Kualitas daging dipengaruhi oleh empat faktor utama yaitu jumlah mikroba, komposisi kimia, sifat fisik, dan nilai pemuas (*eating quality*). Dari keempat faktor ini, konsumen biasanya lebih memperhatikan sifat fisik dan *eating quality* daripada yang lainnya. *Eating quality* sangat tergantung pada kepekaan indera manusia, termasuk penglihatan, penciuman, pengecap, dan perabaan, karena mencakup respons terhadap warna, aroma, rasa, tekstur, serta keseluruhan pengalaman makan daging (Agus Setiadi, Titik Ekowati, 2020).

Sistem pendeteksi tingkat kesegaran daging ayam menjadi solusi penting untuk meningkatkan kualitas dan keamanan pangan. Teknologi canggih dan

metode ilmiah seperti *Convolutional Neural Network* (CNN) dapat membantu meminimalisir risiko penyakit bagi konsumen dan meningkatkan efisiensi dalam rantai pasokan makanan.

Secara teknis, CNN adalah sebuah arsitektur yang dapat dilatih dan terdiri dari beberapa tahap. Input CNN berupa citra, dan proses mengubah citra menjadi fitur yang dapat dipahami oleh jaringan merupakan aspek yang membedakan CNN dari jenis jaringan saraf lainnya. Arsitektur ini melibatkan lapisan konvolusi untuk ekstraksi fitur, lapisan pooling untuk reduksi dimensi, serta lapisan fully connected untuk klasifikasi. Dengan struktur ini, CNN mampu menangkap pola dan fitur yang kompleks dalam citra, sehingga sangat efektif dalam tugas-tugas seperti pengenalan gambar dan deteksi objek (Wulandari et al., 2020).

Penelitian sebelumnya dilakukan oleh Alfinas Agung Mujiono, Kartini, dan Eva Yulia Puspaningrum pada tahun 2024 mengkaji kondisi kesegaran daging ayam menggunakan metode CNN dengan judul “Implementasi Model Hybrid CNN-SVM Pada Klasifikasi Kondisi Kesegaran Daging Ayam”, proses klasifikasi kesegaran daging ayam dilakukan dengan mengintegrasikan dua algoritma pembelajaran mesin: CNN untuk ekstraksi fitur dan SVM untuk klasifikasi. Tujuan dari penelitian ini adalah mengembangkan model *hybrid* yang dapat mengklasifikasikan dua label berdasarkan pra-pemrosesan gambar, ekstraksi fitur, dan klasifikasi citra daging ayam. Model dengan *learning rate* 0.00001 mencapai akurasi tertinggi sebesar 95%, dengan presisi 95%, recall 94.8%, dan f1-score 94.9%. Sebaliknya, model yang paling seimbang diperoleh dengan *learning rate* 0.000001, menghasilkan akurasi 90%, presisi 90.1%, *recall* 90.1%, dan *f1-score* 90.1% (Agung Mujiono et al., 2024).

Penelitian sebelumnya juga dilakukan oleh Rafi Albert Abiyasa, Andrijani Sumarahinsi, dan Rifki Hari Romadhon pada tahun 2023 dengan judul “Implementasi Pengolahan Citra HSV Secara Real-time Sebagai Klasifikasi Tingkat Kesegaran Daging Ayam Potong Dengan Metode KNN”. Penelitian ini bertujuan untuk mendeteksi tingkat kesegaran daging ayam menggunakan pengolahan citra digital dan metode KNN dengan klasifikasi menjadi tiga kategori yaitu segar, kurang segar, dan busuk. Pembaruan dalam penelitian ini mencakup penggunaan sistem operasi yang diproses melalui Raspberry Pi sebagai pengelola data sistem berbasis Linux, dengan bahasa pemrograman Python untuk mengakses data dari kamera yang berfungsi menangkap citra daging ayam (Abiyasa & Romadhon, 2023).

Penelitian sebelumnya juga dilakukan Cakra, Syafruddin Syarif, Hamdan Gani, Andi Patombongi, dan Andi Muh. Islah pada tahun 2022 dengan judul “Analisis Kesegaran Ikan Mujair Dan Ikan Nila Dengan Metode *Convolutional Neural Network*” Berdasarkan hasil penelitian, ikan mujair yang dianalisis menggunakan 451 citra menunjukkan bahwa arsitektur VGG memberikan kinerja terbaik dibandingkan arsitektur lainnya, dengan tingkat akurasi klasifikasi mencapai 73%. Sementara itu, untuk ikan nila yang menggunakan 574 citra, arsitektur VGG juga menunjukkan kinerja lebih baik dibandingkan arsitektur lainnya, dengan akurasi klasifikasi mencapai 57,9%. Temuan ini mengindikasikan bahwa arsitektur VGG sangat efektif dalam pengenalan jenis ikan berdasarkan citra, meskipun tingkat akurasinya bervariasi antar jenis ikan (Cakra et al., 2022).

Dengan memanfaatkan teknologi komputer dan pemrosesan citra, proses pengklasifikasian kesegaran daging ayam dapat dilakukan dengan lebih mudah dan cepat, bahkan oleh individu yang bukan tenaga ahli. Oleh karena itu, penelitian ini bertujuan untuk merancang dan mengimplementasikan sistem deteksi tingkat kesegaran daging ayam berdasarkan parameter warna daging menggunakan pemrosesan citra.

## TINJAUAN PUSTAKA

Saat ini, banyak sistem telah berkembang yang memanfaatkan fitur pengenalan wajah, seperti sistem akses keamanan dan kontrol. Pemrosesan citra dan visi komputer adalah inovasi dalam bidang komputer yang dirancang untuk mendekati kemampuan visual manusia. Pemrosesan citra adalah teknologi yang digunakan untuk menangani masalah terkait gambar, sedangkan visi komputer bertugas membuat keputusan mengenai objek fisik nyata yang diperoleh dari perangkat atau sensor. Visi komputer memungkinkan komputer untuk mengenali citra seperti halnya manusia, salah satunya dalam aplikasi pengenalan wajah (Susim & Darujati, 2021)

Pengolahan citra digital bertujuan utama untuk mengolah dan memanipulasi gambar digital agar sesuai dengan berbagai kebutuhan dan aplikasi. Proses ini mencakup serangkaian operasi yang dirancang untuk meningkatkan kualitas visual gambar, mengekstraksi informasi penting yang terkandung di dalamnya, serta mempermudah interpretasi atau pemanfaatannya dalam berbagai konteks, seperti analisis, pemantauan, dan pengenalan pola. Operasi-operasi ini tidak hanya berfokus pada peningkatan visualisasi tetapi juga pada penyiapan citra untuk langkah-langkah analitis lebih lanjut, termasuk segmentasi, pengenalan objek, dan

pengolahan lanjutan yang dapat mendukung berbagai bidang, seperti medis, industri, keamanan, dan hiburan (Dijaya, 2023).

Format .JPEG, atau *Joint Photographic Experts Group*, merupakan standar kompresi internasional yang dibuat untuk mendukung penggunaan gambar digital, baik berwarna maupun grayscale, serta grafis dengan kualitas tinggi di perangkat digital. Diciptakan sejak tahun 1982, .JPEG menjadi salah satu format gambar digital yang paling umum digunakan. Sebaliknya, PNG, atau *Portable Network Graphics*, adalah format gambar digital yang memakai kompresi lossless. Format ini mendukung transparansi dan portabilitas, serta mampu menangani warna hingga 48 bit, sehingga sangat cocok untuk aplikasi yang membutuhkan kualitas gambar tinggi dan fitur transparansi (Imanudin et al., 2023).

Ayam merupakan salah satu jenis unggas yang sudah sangat dikenal oleh masyarakat. Biasanya, ayam yang digunakan untuk konsumsi adalah ayam potong. Ada beberapa hal yang perlu diperhatikan saat memilih daging ayam segar, seperti warna daging yang putih kekuningan, warna lemak yang juga putih kekuningan dan tersebar merata di bawah kulit, aroma yang segar, serta tekstur daging yang elastis. Selain itu, penting untuk memastikan tidak ada tanda-tanda memar atau indikasi lain yang mencurigakan pada daging ayam tersebut (Laia et al., 2021).

*Convolutional Neural Network* (CNN) adalah salah satu jenis algoritma dalam *Deep Learning* yang mampu menganalisis dataset besar dengan akurasi tinggi. CNN adalah model yang efektif untuk pengenalan aktivitas manusia (HAR), menggunakan teknik jaringan saraf yang kuat, yaitu jaringan saraf konvolusional, untuk memodelkan fitur secara efisien (Hakim, 2021).

*Convolutional Layer* adalah elemen utama dalam CNN yang berfungsi langsung dengan input. Layer ini terdiri dari sejumlah filter yang digunakan untuk mengekstrak fitur dari input yang diberikan. Filter ini ditentukan oleh dimensi lebar dan tinggi. Misalnya, jika layer pertama disebut *convolutional layer* 5x5x3, berarti filternya memiliki lebar 5 piksel, tinggi 5 piksel, dan 3 kanal, sesuai dengan struktur gambar. Setiap filter digeser melintasi seluruh bagian citra, dan setiap kali terjadi pergeseran, dilakukan operasi "*dot product*" antara input dan nilai filter untuk menghasilkan *output* yang disebut *activation map* atau *feature map*. Proses ini memungkinkan CNN untuk mendeteksi pola-pola penting dalam gambar dan membangun representasi fitur yang diperlukan untuk

tugas klasifikasi atau deteksi selanjutnya (Prastowo, 2021).

*Pooling Layer* adalah lapisan yang menggunakan *Feature Map* dari *Convolution Layer* sebagai input untuk mengecilkan ukuran spasial, dengan tujuan mengurangi kebutuhan sumber daya komputasi dalam pengolahan data. Ada dua jenis pooling yang umum digunakan, yaitu *Max Pooling* dan *Average Pooling*. Perbedaan utama di antara keduanya terletak pada cara memilih nilai dari area tertentu pada *feature map* (Herianto et al., 2023).

*Fully connected layer* adalah jenis jaringan saraf yang mirip dengan jaringan saraf tradisional. Pada tahap ini, klasifikasi gambar dilakukan dengan memanfaatkan fitur yang telah diekstrak selama proses konvolusi. Setiap nilai dari *feature map* diubah menjadi input untuk *layer* ini, yang mengakibatkan jumlah bobot yang sangat besar karena setiap neuron terhubung dengan *neuron* di *layer* berikutnya. Karena itu, sebagian besar parameter dalam CNN terdapat pada *fully connected layers*. Untuk melatih bobot-bobot ini, metode *backpropagation* digunakan, yang memungkinkan model untuk menyesuaikan bobot berdasarkan kesalahan yang dihasilkan selama proses pelatihan (Yulianto et al., 2023).

ReLU (*Rectified Linear Unit*) adalah fungsi aktivasi non-linear yang beroperasi pada level piksel. Fungsi ini mengeluarkan nilai input  $x$  jika  $x$  positif, dan 0 jika  $x$  negatif. Secara matematis, ReLU dinyatakan sebagai  $g(x) = \max(0, x)$  (Herianto et al., 2023).

*Flatten* adalah fungsi yang mengubah gambar berbentuk multidimensional array dari hasil konvolusi menjadi array satu dimensi. Fungsi ini mempersiapkan data untuk proses klasifikasi dengan menyusun semua elemen menjadi vektor satu dimensi, yang dapat digunakan sebagai input untuk *layer fully connected* dalam jaringan saraf (Herianto et al., 2023).

*Softmax* adalah fungsi aktivasi yang digunakan untuk klasifikasi linear dengan menghasilkan probabilitas untuk setiap kelas. *Output* dari fungsi ini berada dalam rentang 0 hingga 1, di mana jumlah total probabilitas untuk semua kelas dalam satu objek output adalah 1 (Herianto et al., 2023). Keuntungan utama dari *Softmax* adalah kemampuannya untuk mengonversi nilai-nilai *output* menjadi probabilitas yang dapat dipahami, serta kemampuannya untuk digunakan dalam model regresi logistik klasifikasi ganda (Kholik, 2021).

## METODE PENELITIAN

Perangkat keras (*hardware*) adalah komponen fisik dari komputer yang dapat disentuh dan dilihat langsung. Ini meliputi bagian-bagian yang digunakan untuk input, pengolahan data, dan output. Perangkat keras penting untuk menjalankan perintah yang diprogram dan mendukung operasi sistem secara keseluruhan.

1. Advan Workplus
2. Processor AMD Ryzen 5 6600H, RAM 16,00 GB, SSD 1TB
3. *Smartphone* Android Realme 10, Kamera 50MP.

Perangkat lunak (*software*) adalah kumpulan data elektronik yang disimpan dan dikelola oleh komputer. Data ini berupa program atau instruksi yang mengarahkan perangkat keras untuk menjalankan perintah. Perangkat lunak berfungsi sebagai penerjemah perintah dari pengguna, memungkinkan komputer untuk memproses dan melaksanakan instruksi yang diberikan. Selain itu, perangkat lunak juga merupakan elemen penting dalam merancang dan membangun sebuah sistem. Fungsinya mencakup pengolahan data. Perangkat lunak yang digunakan dalam perancangan sistem ini adalah:

1. Sistem Operasi *Microsoft Windows 11 Pro, 64 bit*
2. *Google Chrome, Google Colaboratory*
3. *Android Studio*.

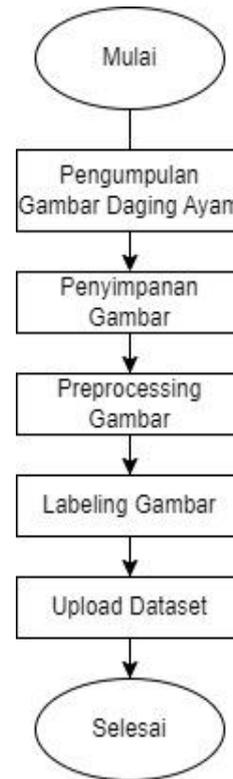
Pengumpulan data dilakukan dengan cara mengambil sampel yang representatif dari berbagai tingkat kesegaran daging. Dalam penelitian ini, citra atau gambar sampel diambil langsung dari pasar dan melibatkan beberapa kondisi, yaitu daging ayam dalam keadaan segar, kurang segar, dan busuk. Daging ayam segar diambil segera setelah dibeli, sementara daging ayam yang kurang segar dan busuk diperoleh dengan membiarkan daging ayam mengalami proses pembusukan. Proses ini memastikan bahwa sampel yang dikumpulkan mencakup berbagai kondisi kesegaran, memungkinkan analisis yang lebih menyeluruh dalam penelitian ini.

### Flowchart Pengumpulan Data

*Flowchart* pengumpulan data ini berisi Langkah-langkah yang dilakukan saat mengumpulkan data.

Adapun tahap-tahap yang dilakukan pada gambar 1 yaitu melakukan pembelian daging ayam di pasar. Lakukan pengambilan gambar daging ayam yang dibeli di pasar menggunakan kamera *smartphone* android. Simpan gambar di penyimpanan lokal. *Preprocessing* gambar berisi *resizing* dan penghapusan *background*. Setiap gambar yang telah dipreproses

diberikan label sesuai dengan tingkat kesegaran daging ayam (segar, kurang segar, busuk). *Upload* dataset ke *Kaggle*.

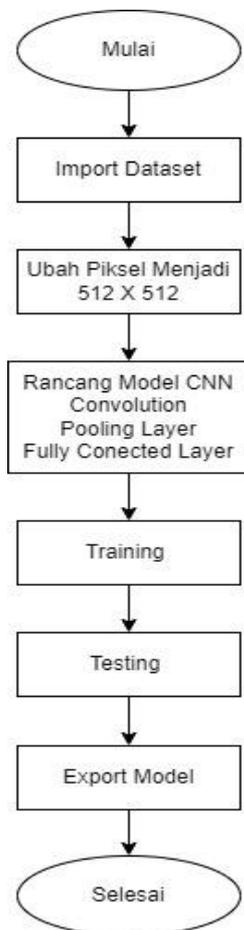


Gambar 1. Flowchart Pengumpulan Data

### Flowchart Pembuatan Model CNN

*Flowchart* pembuatan model CNN ini berisi Langkah-langkah yang dilakukan saat pembuatan model CNN.

Adapun tahap-tahap yang dilakukan pada gambar 2 yaitu melakukan pembuatan model di *google colaboratory*. *Import* dataset dari *Kaggle*. Ubah nilai *input* piksel menjadi 512x512. Rancang model CNN, pada *layer convolution* menggunakan filter 3x3 sebanyak 32 buah filter, menggunakan aktivasi *ReLU*, *zero padding* dan menggunakan *stride* atau pergeseran filter sebanyak 1, *pooling layer* menggunakan *average pooling*, selanjutnya menggunakan fungsi aktivasi *flatten*, dan terakhir untuk proses klasifikasi pada *fully connected layer* menggunakan aktivasi *softmax*. *Training* untuk melihat nilai akurasi dan nilai kesalahan. *Testing* untuk mengevaluasi kinerja sistem. *Export* model kedalam *.tflite* untuk di implementasikan kedalam sistem android.

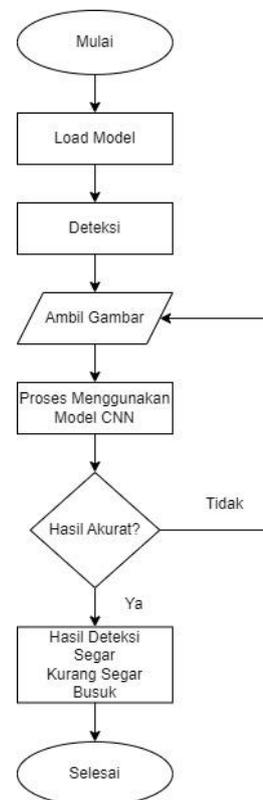


Gambar 2. Flowchart Pembuatan Model CNN

### Flowchart Aplikasi Android

Flowchart aplikasi android ini berisi langkah-langkah yang dilakukan saat penggunaan aplikasi android di *smartphone*.

Adapun tahap-tahap yang dilakukan pada gambar 2 yaitu Mulai dengan masuk kedalam aplikasi android. Setelah masuk kedalam aplikasi android akan otomatis memuat model CNN yang sudah dibuat sebelumnya pada *google collaboratory*. Setelah selesai memuat model akan masuk ke *dashboard* atau halaman utama, selanjutnya *user* memilih menu deteksi untuk melakukan pendeteksian. Arahkan kamera ke potongan daging ayam yang akan di deteksi. Proses pendeteksian akan dilakukan menggunakan model CNN yang telah dibuat. Jika hasil sesuai yang di inginkan akan keluar hasil yang sebenarnya, jika tidak sesuai silahkan kembali mengambil ulang gambar. Hasil akan berupa persentase dimana hasil yang paling tinggi atau mendekati akan berada di posisi paling atas.



Gambar 3. Flowchart Aplikasi Android

## HASIL DAN PEMBAHASAN

Penelitian ini mengevaluasi penggunaan *metode Convolutional Neural Network* untuk mendeteksi tingkat kesegaran potongan daging ayam. Hasil dari penelitian ini akan menghasilkan sebuah aplikasi Android yang dapat mendeteksi tingkat kesegaran daging ayam secara *real-time*.

### Pembuatan Model

#### Import Dataset

```
!kaggle datasets download rayhannaturizal/fixdata
```

```
Dataset URL: https://www.kaggle.com/datasets/rayhannaturizal/fixdata
License(s): unknown
Downloading fixdata.zip to /content
91% 12.0M/13.1M [00:01<00:00, 13.4MB/s]
100% 13.1M/13.1M [00:01<00:00, 8.30MB/s]
```

Gambar 4. Sourcecode Import Dataset

#### Rancangan Model CNN

Perancangan model pada metode CNN digunakan untuk menentukan jumlah filter, ukuran *stride*, fungsi aktivasi dan lain sebagainya pada metode CNN.

```
model = tf.keras.Sequential([
    base_model,
    tf.keras.layers.Conv2D(32, 3, padding='same', strides=1, activation='relu'),
    tf.keras.layers.AveragePooling2D(32, 2, strides=2),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(3, activation='softmax')
])
```

Gambar 5. Sourcecode Rancangan Model CNN

Gambar berikut menunjukkan bahwa pada proses *convolution* digunakan 32 filter berukuran 3x3, menghasilkan 32 *feature map* baru dengan *padding same*, *stride* 1, dan fungsi aktivasi ReLU. Kemudian, pada lapisan *pooling* menggunakan *average pooling* dengan 32 filter berukuran 2x2 dan *stride* 2. *Output* dari *pooling* diubah menjadi vektor dengan fungsi aktivasi *flatten*. Pada lapisan *fully connected layer*, digunakan *dense* dengan 3 unit untuk klasifikasi tingkat kesegaran daging ayam segar, kurang segar, dan busuk menggunakan fungsi aktivasi *softmax*.

**Training Model**

```
[ ] from keras.callbacks import EarlyStopping, ModelCheckpoint

history = model.fit(train_generator,
                    epochs=50,
                    validation_data=val_generator)
```

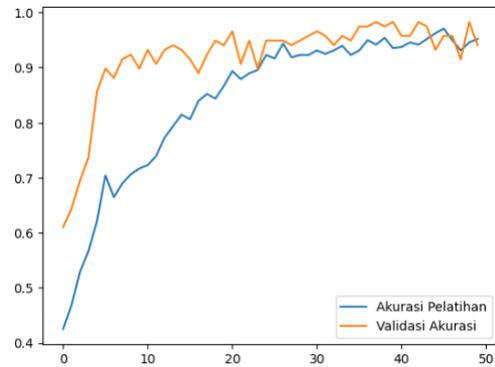
Gambar 6. Sourcecode Proses Training

Ketika *training* dimulai, akan muncul terminal window yang menampilkan proses *training* seperti pada gambar berikut.

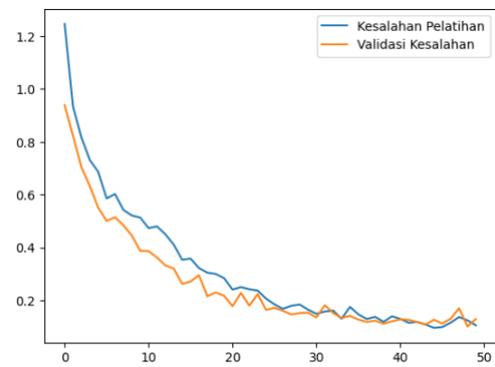
```
Epoch 48/50 ----- 7s 328ms/step - accuracy: 0.9298 - loss: 0.1414 - val_accuracy: 0.9831 - val_loss: 0.1201
Epoch 41/50 ----- 6s 292ms/step - accuracy: 0.9386 - loss: 0.1385 - val_accuracy: 0.9576 - val_loss: 0.1274
Epoch 42/50 ----- 6s 323ms/step - accuracy: 0.9483 - loss: 0.1241 - val_accuracy: 0.9576 - val_loss: 0.1258
Epoch 43/50 ----- 6s 281ms/step - accuracy: 0.9426 - loss: 0.1153 - val_accuracy: 0.9831 - val_loss: 0.1172
Epoch 44/50 ----- 6s 279ms/step - accuracy: 0.9556 - loss: 0.1024 - val_accuracy: 0.9746 - val_loss: 0.1075
Epoch 45/50 ----- 6s 311ms/step - accuracy: 0.9706 - loss: 0.0902 - val_accuracy: 0.9322 - val_loss: 0.1263
Epoch 46/50 ----- 6s 389ms/step - accuracy: 0.9753 - loss: 0.0887 - val_accuracy: 0.9576 - val_loss: 0.1115
Epoch 47/50 ----- 6s 277ms/step - accuracy: 0.9406 - loss: 0.1246 - val_accuracy: 0.9576 - val_loss: 0.1303
Epoch 48/50 ----- 7s 328ms/step - accuracy: 0.9371 - loss: 0.1302 - val_accuracy: 0.9353 - val_loss: 0.1695
Epoch 49/50 ----- 6s 276ms/step - accuracy: 0.9312 - loss: 0.1360 - val_accuracy: 0.9831 - val_loss: 0.1006
Epoch 50/50 ----- 6s 276ms/step - accuracy: 0.9617 - loss: 0.0994 - val_accuracy: 0.9487 - val_loss: 0.1285
```

Gambar 7. Proses Training

Setelah proses data *training* selesai dilakukan sebanyak 50 *epoch*, dibutuhkan 2 buah grafik untuk mengetahui proses naik turun akurasi pada proses *training*.



Gambar 8. Grafik Akurasi Pelatihan dan Validasi Akurasi



Gambar 9. Grafik Kesalahan Pelatihan dan Validasi Kesalahan

**Testing Model**

Setelah proses *training* perlu dilakukan *testing* untuk menguji model apakah mampu melakukan prediksi dengan baik atau tidak.

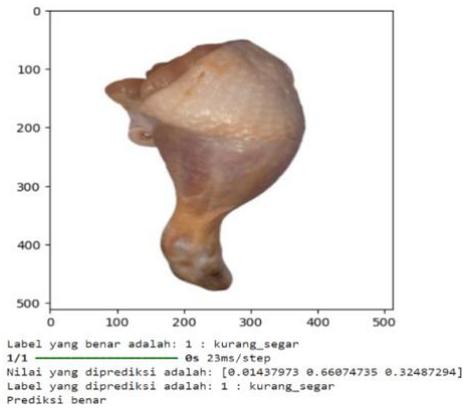
```
plt.imshow(X_test[n])
plt.show()

true_label = np.argmax(y_test2,axis=1)[n]
print("Label yang benar adalah:",true_label,",",labels[true_label])
prediction = model.predict(X_test[n][np.newaxis,...])[0]
print("Nilai yang diprediksi adalah:",prediction)
predicted_label = np.argmax(prediction)
print("Label yang diprediksi adalah:",predicted_label,",",labels[predicted_label])

if true_label == predicted_label:
    print("Prediksi benar")
else:
    print("Prediksi salah")
```

Gambar 10. Sourcecode Proses Testing

Setelah proses testing dilakukan akan muncul sebuah label prediksi seperti gambar berikut:



Gambar 11. Hasil Proses Testing

**Export Model**

Setelah hasil testing maupun training model sesuai yang di inginkan, kemudian model di export ke dalam bentuk file TFLite (.tflite) untuk penerapan secara *real-time* pada aplikasi android.

```
import tensorflow as tf

try:
    # Convert model to TensorFlow Lite format
    converter = tf.lite.TFLiteConverter.from_keras_model(model)
    tflite_model = converter.convert()

    # Save the model
    with open('model.tflite', 'wb') as f:
        f.write(tflite_model)

except Exception as e:
    print("Error saving model:", e)
    # Handle the error, e.g., retry, log, or notify user
```

Gambar 12. Sourcecode Proses Export Model

**Perancangan Interface**

**Halaman Load Model**

Halaman *load* model adalah proses memuat model yang telah dibuat sebelumnya pada *google colab* dan disimpan dalam bentuk file-file dan di implementasikan ke aplikasi android.



Gambar 13. Tampilan Load Model

**Halaman Dashboard**



Gambar 14. Tampilan Load Model

**Halaman Deteksi**

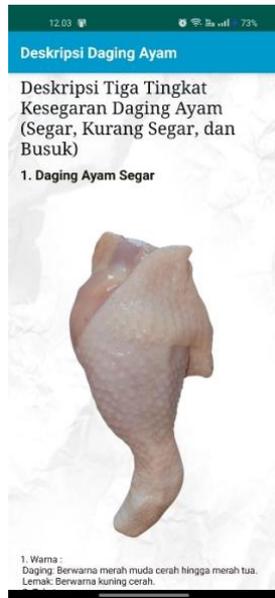
Halaman deteksi akan menampilkan kamera belakang pada smartphone android user yang berfungsi untuk mendeteksi tingkat kesegaran pada potongan daging ayam dan akan mengeluarkan hasil secara *real-time* berbentuk persentase dengan urutan hasil yang paling mendekati ada di posisi pertama atau paling atas pada hasil.



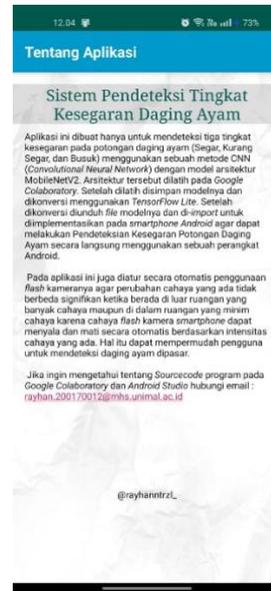
Gambar 15. Tampilan Deteksi

**Halaman Deskripsi**

Halaman deskripsi berisikan informasi tentang ciri-ciri daging ayam dari mulai daging ayam yang segar hingga busuk.



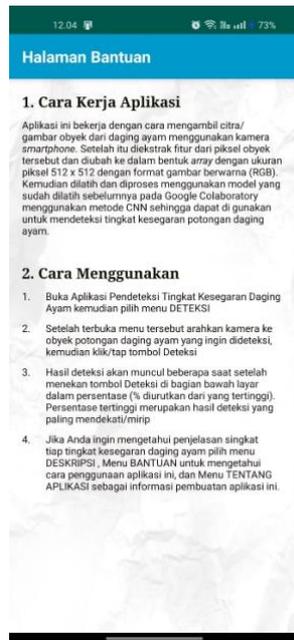
Gambar 16. Tampilan Deskripsi



Gambar 18. Tampilan Tentang Aplikasi

**Halaman Bantuan**

Halaman bantuan ini berisikan tentang cara penggunaan aplikasi, dan fitur-fitur yang terdapat pada aplikasi.



Gambar 17. Tampilan Bantuan

**Pengujian Aplikasi Android**

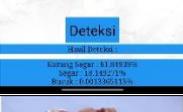
Pengujian ini dilakukan untuk menguji potongan daging ayam dengan berbagai kondisi yang berbeda. Pengujian ini dilakukan menggunakan smartphone android realme 10 dengan resolusi kamera sebesar 50MP. Kondisi cahaya saat melakukan pengujian juga berbeda-beda, tetapi pada aplikasi android ini menggunakan *flash* yang otomatis aktif apabila pengujian dilakukan pada kondisi cahaya yang minim. Pengujian ini dilakukan untuk mengetahui seberapa akurat model yang di implementasikan pada perangkat android untuk digunakan mendeteksi tingkat kesegaran pada potongan daging ayam.

Tabel 1. Pengujian Aplikasi Android

No	Pengujian	Kondisi	Hasil Deteksi
1		Segar	Segar 99%

**Halaman Tentang Aplikasi**

Halaman tentang aplikasi berisi informasi tentang pembuatan aplikasi.

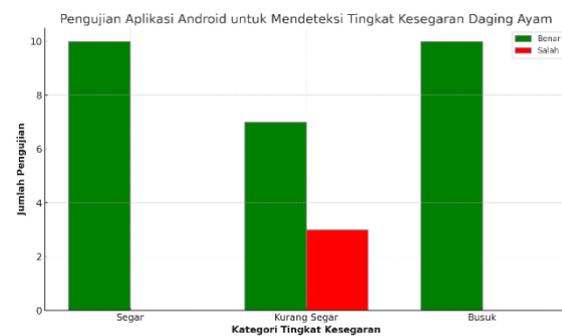
2	  Segar : 100.0% Kurang Segar : 0.000000% Busuk : 0.000000%	Segar	Segar 100%	8	  Segar : 100.0% Kurang Segar : 0.000000% Busuk : 0.000000%	Segar	Segar 100%
3	  Segar : 99.9999% Kurang Segar : 0.000001% Busuk : 0.000000%	Segar	Segar 99%	9	  Segar : 83.0000% Kurang Segar : 16.0000% Busuk : 0.000000%	Segar	Segar 83%
4	  Segar : 99.9999% Kurang Segar : 0.000001% Busuk : 0.000000%	Segar	Segar 99%	10	  Segar : 100.0000% Kurang Segar : 0.000000% Busuk : 0.000000%	Segar	Segar 100%
5	  Segar : 99.9999% Kurang Segar : 0.000001% Busuk : 0.000000%	Segar	Segar 99%	11	  Kurang Segar : 81.0000% Segar : 19.0000% Busuk : 0.000000%	Kurang Segar	Kurang Segar 81%
6	  Segar : 99.9999% Kurang Segar : 0.000001% Busuk : 0.000000%	Segar	Segar 99%	12	  Kurang Segar : 76.0000% Segar : 24.0000% Busuk : 0.000000%	Kurang Segar	Busuk 76%
7	  Segar : 100.0% Kurang Segar : 0.000000% Busuk : 0.000000%	Segar	Segar 100%	13	  Segar : 99.9999% Kurang Segar : 0.000001% Busuk : 0.000000%	Kurang Segar	Segar 99%

14	 <b>Deteksi</b> Hasil Deteksi: Kurang Segar : 61,389579% Segar : 32,904727% Busuk : 10,685694%	Kurang Segar	Kurang Segar 63%	20	 <b>Deteksi</b> Hasil Deteksi: Kurang Segar : 94,1294% Segar : 3,03771441% Busuk : 0,83288559%	Kurang Segar	Kurang Segar 99%
15	 <b>Deteksi</b> Hasil Deteksi: Kurang Segar : 58,225776% Segar : 32,612542% Busuk : 9,16169682%	Kurang Segar	Kurang Segar 68%	21	 <b>Deteksi</b> Hasil Deteksi: Busuk : 94,982857% Kurang Segar : 1,80818827% Segar : 1,43896315%	Busuk	Busuk 99%
16	 <b>Deteksi</b> Hasil Deteksi: Kurang Segar : 75,862221% Segar : 17,432279% Busuk : 5,704499%	Kurang Segar	Kurang Segar 77%	22	 <b>Deteksi</b> Hasil Deteksi: Busuk : 94,892857% Kurang Segar : 0,21188277% Segar : 4,27764474%	Busuk	Busuk 99%
17	 <b>Deteksi</b> Hasil Deteksi: Kurang Segar : 72,80539% Segar : 27,193606% Busuk : 0,1149401%	Kurang Segar	Kurang Segar 72%	23	 <b>Deteksi</b> Hasil Deteksi: Busuk : 99,982857% Kurang Segar : 0,0151188277% Segar : 1,0020226%	Busuk	Busuk 99%
18	 <b>Deteksi</b> Hasil Deteksi: Kurang Segar : 88,28839% Segar : 9,111127% Busuk : 2,59948274%	Kurang Segar	Kurang Segar 96%	24	 <b>Deteksi</b> Hasil Deteksi: Busuk : 94,76514% Kurang Segar : 0,21848961% Segar : 4,31635899%	Busuk	Busuk 99%
19	 <b>Deteksi</b> Hasil Deteksi: Segar : 92,28914% Kurang Segar : 0,79533918% Busuk : 0,7155295074%	Kurang Segar	Segar 99%	25	 <b>Deteksi</b> Hasil Deteksi: Busuk : 99,982857% Kurang Segar : 0,0151188277% Segar : 1,0020226%	Busuk	Busuk 99%

26	 	Busuk	Busuk 99%
27	 	Busuk	Busuk 99%
28	 	Busuk	Busuk 80%
29	 	Busuk	Busuk 60%
30	 	Busuk	Busuk 99%

$$\frac{27}{30} \times 100\% = 90\%$$

Setelah melakukan pengujian sebanyak total 30 kali pengujian dan tiap masing-masing kategori di uji sebanyak 10 kali pada aplikasi Android untuk mendeteksi tingkat kesegaran daging ayam dengan rincian 27 hasil deteksi benar dan hasil deteksi yang salah sebanyak 3. Hasilnya menunjukkan bahwa aplikasi mampu mendeteksi dengan benar pada kategori segar dan busuk dalam semua pengujian yang dilakukan. Namun, pada kategori kurang segar, terdapat 3 kesalahan dari 10 pengujian yang dilakukan, dengan tingkat akurasi keseluruhan mencapai 90%.



Gambar 18. Grafik Hasil Pengujian Aplikasi Android

### KESIMPULAN

Penelitian ini berhasil mengembangkan sebuah sistem pendeteksi tingkat kesegaran daging ayam berbasis citra yang diimplementasikan pada platform Android, menggunakan metode *Convolutional Neural Network* (CNN).

Sistem ini mampu mengklasifikasikan tingkat kesegaran daging ayam ke dalam tiga tingkat kesegaran yaitu segar, kurang segar, dan busuk. Penggunaan metode CNN terbukti efektif dalam mengolah citra daging ayam untuk mendeteksi tingkat kesegaran.

Dari 30 kali pengujian, masing-masing kategori diuji 10 kali dengan akurasi total 90%, di mana sistem berhasil 27 kali dan mengalami 3 kesalahan pada kategori kurang segar. Performa sangat baik ditunjukkan pada deteksi kategori segar dan busuk, namun kesalahan pada kategori kurang segar menunjukkan tantangan dalam membedakan visual yang lebih ambigu.

### DAFTAR PUSTAKA

Abiyasa, R. A., & Romadhon, R. H. (2023). Implementasi Pengolahan Citra HSV Secara Real Time Sebagai Klasifikasi Tingkat Kesegaran Daging Ayam Potong Dengan Metode KNN. *Seminar Nasional Teknologi Industri*, 1(1), 1001–1010.

Penelitian serta hasil penelitian ini dilakukan pada agustus 2024. Pengujian ini dilakukan secara real-time dengan cara membeli daging ayam yang masih segar sebagai sampel, dan dilakukan pembusukan untuk menguji sambel daging ayam yang kurang segar dan busuk.

Berdasarkan data hasil pengujian, perhitungan akurasi:

- Agung Mujiono, A., Kartini, K., & Yulia Puspaningrum, E. (2024). IMPLEMENTASI MODEL HYBRID CNN-SVM PADA KLASIFIKASI KONDISI KESEGARAN DAGING AYAM. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 8(1), 756–763. <https://doi.org/10.36040/jati.v8i1.8855>
- Agus Setiadi, Titik Ekowati, K. A. (2020). Analisis Preferensi Konsumen Dalam Membeli Daging Ayam Broiler Di Pasar Tradisional Kota Semarang, Jawa Tengah. *AGROMEDIA: Berkala Ilmiah Ilmu-Ilmu Pertanian*, 38(2), 76–89. <https://doi.org/10.47728/ag.v38i2.287>
- Cakra, C., Syarif, S., Gani, H., & Patombongi, A. (2022). ANALISIS KESEGARAN IKAN MUJAIR DAN IKAN NILA DENGAN METODE CONVOLUTIONAL NEURAL NETWORK. *Simtek : Jurnal Sistem Informasi Dan Teknik Komputer*, 7(2), 74–79. <https://doi.org/10.51876/simtek.v7i2.138>
- Dijaya, R. (2023). Buku Ajar Pengolahan Citra Digital. In *Umsida Press*. Umsida Press. <https://doi.org/10.21070/2023/978-623-464-075-5>
- Hakim, A. A. (2021). Klasifikasi Human Activity Recognition Menggunakan Metode CNN. *Jurnal Repositor*, 3(2). <https://doi.org/10.22219/repositor.v3i2.1265>
- Herianto, Adam Arif Budiman, Linda Nur Afifa, Timor Setiyaningsih, & Tri Amin Ridho. (2023). Membangun Model Pengidentifikasi Kesegaran Daging dengan Metode Jaringan Syaraf Konvolusi (CNN) Jenis Resnet-50. *IKRA-ITH Informatika : Jurnal Komputer Dan Informatika*, 7(3), 113–119. <https://doi.org/10.37817/ikraith-informatika.v7i3.3072>
- Imanudin, R. F. N., Kustiawan, I., & Elvyanti, S. (2023). Steganografi Citra Digital Menggunakan Pendekatan Least Significant Bit dan Discrete Cosine Transform. *Seminar Nasional Teknik Elektro*.
- Kholik, A. (2021). Klasifikasi Menggunakan Convolutional Neural Network (CNN) Pada Tangkapan Layar Instagram. *Jurnal Data Mining Dan Sistem Informasi*, 2(2), 10. <https://doi.org/10.33365/jdmsi.v2i2.1345>
- Laia, M., Hondro, R. K., & Zebua, T. (2021). Implementasi Pengolahan Citra dengan Menggunakan Metode K-Nearest Neighbor Untuk Mengetahui Daging Ayam Busuk dan Daging Ayam Segar. *JURIKOM (Jurnal Riset Komputer)*, 8(2), 39–49. <https://doi.org/http://dx.doi.org/10.30865/jurikom.v8i2.2818>
- Prastowo, E. Y. (2021). Pengenalan Jenis Kayu Berdasarkan Citra Makroskopik Menggunakan Metode Convolutional Neural Network. *Jurnal Teknik Informatika Dan Sistem Informasi*, 7(2), 489–497. <https://doi.org/10.28932/jutisi.v7i2.3706>
- Susim, T., & Darujati, C. (2021). Pengolahan Citra untuk Pengenalan Wajah (Face Recognition) Menggunakan OpenCV. *Jurnal Syntax Admiration*, 2(3), 534–545. <https://doi.org/10.46799/jsa.v2i3.202>
- Wulandari, I., Yasin, H., & Widiharih, T. (2020). Klasifikasi Citra Digital Bumbu dan Rempah Dengan Algoritma Convolutional Neural Network (CNN). *Jurnal Gaussian*, 9(3), 273–282. <https://doi.org/10.14710/j.gauss.v9i3.27416>
- Yulianto, A., Andreas, W., & Sabariman, S. (2023). Perancangan Prototype Brankas Menggunakan Sistem Pengenalan Wajah Dengan Metode Convolutional Neural Network (CNN). *Telcomatics*, 8(1), 10. <https://doi.org/10.37253/telcomatics.v8i1.7852>