

Quameaty: Aplikasi Pendeteksi Kualitas Daging Ayam Mentah Berbasis Pengolahan Citra Menggunakan Model InceptionV3

Amir Husein[#], Abyan Ramzi, Nur Ihsan Muzakki, Rahmawati Hasanah

Jurusan Teknik Elektro, Politeknik Negeri Bandung

Jl. Gegerkalong Hilir Ds. Ciwaruga, Bandung Barat 40559, Jawa Barat, Indonesia

[#]amir.husein.tkom418@polban.ac.id

Abstrak

Dewasa ini, daging ayam telah menjadi sumber protein hewani yang baik untuk dikonsumsi dan mudah untuk didapatkan. Akan tetapi, dalam proses mendapatkannya seringkali ditemukan praktik curang, seperti daging ayam tiren yang tetap dijual, ataupun daging ayam yang telah dicampur dengan daging yang tidak layak jual. Maka dari itu, diperlukan sebuah alat atau aplikasi yang mampu mendeteksi kualitas daging ayam mentah. Tujuan penelitian ini untuk membuat alat yang berguna dalam mendeteksi kualitas daging ayam mentah dengan memanfaatkan pengolahan citra menggunakan model InceptionV3 dan diberi nama Quameaty. Alat ini dikembangkan menggunakan bahasa pemrograman Python. Model InceptionV3 merupakan model pelatihan jaringan saraf konvolusional yang sangat baik dan telah dilatih sebelumnya pada dataset *Common Objects in Context* (COCO) yang berjumlah 328 ribu gambar dengan 81 kelas yang berbeda. Model ini memiliki tingkat akurasi yang sangat tinggi sebagai *pre-trained model* dengan nilai *top 5 accuracy* sebesar 93,3% dan waktu komputasi yang relatif cepat apabila dibandingkan dengan model pendahulunya. Model pelatihan yang dihasilkan ditanam pada aplikasi Android yang mana dapat dengan mudah dan cenderung fleksibel untuk digunakan dalam mendeteksi kualitas daging ayam mentah. Hasil penelitian dibagi menjadi dua kelas yaitu segar dan busuk, serta menunjukkan bahwa prediksi kualitas daging ayam mentah berjalan dengan baik dengan nilai metrik pengujian yang telah mencapai lebih dari 90% pada dua nilai *threshold*, yaitu 50% dan 75%.

Kata kunci: daging, pengolahan citra, jaringan saraf konvolusional, InceptionV3, Android

Abstract

Nowadays, chicken meat has become a good source of animal protein for consumption and easy to obtain. However, in the process of obtaining it, fraudulent practices are often found, such as tiren chicken meat that is still being sold, or chicken meat that has been mixed with meat that is not worth selling. Therefore, we need a tool or application that is able to detect the quality of raw chicken meat. The purpose of this study is to create a tool that is useful in detecting the quality of raw chicken meat by utilizing image processing using the InceptionV3 model and named Quameaty. This tool was developed using the Python programming language. The InceptionV3 model is an excellent convolutional neural network training model and has been pre-trained on the *Common Objects in Context* (COCO) dataset of 328,000 images with 81 different classes. This model has a very high level of accuracy as a pre-trained model with a top 5 accuracy value of 93.3% and a relatively fast computation time when compared to its predecessor model. The resulting training model is embedded in an Android application which can be easily and tends to be flexible to be used in detecting the quality of raw chicken meat. The results of the study were divided into two classes, namely fresh and rotten, and showed that the prediction of the quality of raw chicken meat went well with the test metric values that had reached more than 90% at two threshold values, namely 50% and 75%.

Keywords: meat, image processing, convolutional neural network, InceptionV3, Android

I. PENDAHULUAN

Daging merupakan sumber protein utama bagi masyarakat, baik daging sapi, kambing, ayam dan

lainnya. Seiring berjalannya waktu dan pertumbuhan penduduk yang kian meningkat maka kebutuhan daging ayam juga ikut meningkat. Meningkatnya jumlah konsumen tidak serta merta

meningkat pula jumlah penyuplai daging. Hal ini dimanfaatkan oleh sebagian kelompok manusia untuk menggunakan langkah-langkah licik seperti mencampur daging layak konsumsi dengan daging busuk, menjual daging yang sudah tidak layak konsumsi bahkan mencampurkan zat berbahaya agar tahan lama seperti formalin. Hal ini dilakukan agar keuntungan yang didapat oleh pedagang semakin besar, tanpa memperdulikan keselamatan orang lain [1].

Telah ditemukan beberapa solusi untuk mendeteksi kualitas daging berbasis *image processing* diantaranya yaitu menggunakan metode *Hyperspectral Imaging* yang mana pada prosesnya menganalisis perubahan dan komposisi warna antar region, akan tetapi masih terkendala pada waktu pemrosesan yang relatif lama [2]-[4]. Selain itu terdapat juga solusi dengan metode *K-Nearest Neighbors* (KNN), tetapi memiliki kelemahan karena tidak dilakukannya proses augmentasi gambar sehingga bentuk lain dari sebuah data latih tidak diketahui [5]. Kemudian terdapat penelitian untuk membedakan beberapa jenis daging hewan mentah dengan menganalisis ekstrak asam amino dari tiap daging, yang mana proses ini membutuhkan biaya yang besar juga proses identifikasi secara teliti dan menyeluruh pada setiap fase yang dilakukan [6]. Proses monitoring pun dilakukan pada penelitian yang dilakukan oleh Viancy, dkk dengan memanfaatkan *machine learning* dan IoT, dengan penggunaan sensor elektronik yang memiliki kelemahan berupa respon sensor akan semakin lama seiring dengan meningkatnya waktu penggunaan alat [7]. Penggunaan metode *image processing* terbukti baik dalam melakukan pendeteksian apabila dibandingkan dengan metode *electronic nose*, dikarenakan penggunaan metode *electronic nose* memerlukan perangkat elektronik yang dirancang serta penggunaannya yang tidak efisien dan nilai akurasi yang cenderung berubah-ubah menyesuaikan dengan kondisi lingkungan sekitar terutama dari faktor suhu [8].

Untuk mengatasi permasalahan tersebut, penulis memberi solusi yaitu aplikasi untuk mendeteksi jenis dan kondisi daging. Kelebihan dari sistem ini adalah yang digunakan mudah untuk berpindah tempat (*portable*) dan data hasil deteksi tersebut dapat dipantau langsung melalui *smartphone* android. Pada dasarnya alat ini dirancang untuk mendeteksi objek visual dari daging ayam. Alat tersebut berupa aplikasi android yang di dalamnya telah ditanam model *Convolutional Neural Network* (CNN) yang dirancang mandiri juga memanfaatkan *transfer learning* dari *pre-trained* model InceptionV3 untuk melakukan pendeteksian

kesegaran daging dari objek visual daging itu sendiri yang ditangkap dengan kamera ponsel android yang dilengkapi lensa makro. CNN sendiri merupakan sebuah metode pengolahan citra yang memanfaatkan beberapa lapisan pemrosesan didalamnya dengan tujuan untuk mempelajari detail dan keunikan dari tiap data gambar yang dilatih [9]. Alat ini akan dirancang dari awal, mulai dari proses pengumpulan data sampel untuk dilakukan proses pelatihan hingga proses integrasi dengan *smartphone* android. Penggunaan model Inceptionv3 bertujuan untuk mempercepat proses pelatihan dan juga meningkatkan akurasi yang ada. Hal ini dikarenakan Inceptionv3 adalah sebuah model yang dirancang oleh Google dan telah mengenali banyak kelas yang berbeda dari proses pelatihan pada dataset ImageNet dan COCO. Inceptionv3 juga merupakan model yang memiliki 6,4 juta parameter didalamnya dari lapisan konvolusi hingga *maxpool*, tetapi jumlah ini lebih sedikit ketimbang para pendahulunya. Jumlah parameter yang tidak sebanyak model lainnya ini akan mempercepat proses pelatihan tanpa mengurangi kualitas pengukuran dan pengujian yang ada [10].

II. METODE PENELITIAN

A. Persiapan Data

Data yang dibutuhkan adalah berupa gambar daging ayam yang dibagi menjadi dua label yaitu segar dan busuk dengan porsi jumlah data yang sama antar kedua label tersebut. Proses pembusukan daging ayam dilakukan dengan cara menyimpan daging ayam pada suhu ruangan selama kurang lebih 3 hari sampai daging ayam mentah mengeluarkan cairan berlebih.

Proses pengambilan data gambar daging ayam segar dan busuk dilakukan dengan proses *take video* yang kemudian akan dipisahkan untuk setiap *frame* nya sebagai data yang akan diolah dalam proses pelatihan.

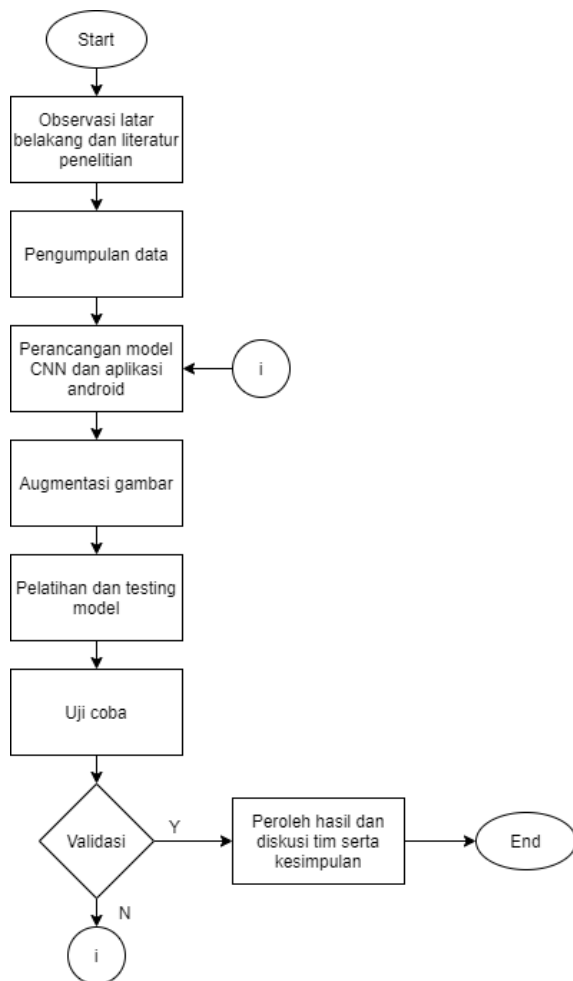
B. Perancangan

Perancangan dibagi menjadi 5 tahap, yaitu pada tahap pertama, proses pengumpulan data awal yang kemudian akan dipisahkan antara data latih dan *testing*. Tahap kedua yaitu penentuan model CNN yang akan digunakan dari perancangan *layer* input hingga *layer* output serta melakukan *transfer learning* menggunakan *pre-trained* model InceptionV3 juga penentuan parameter augmentasi gambar yang akan digunakan. Tahap ketiga adalah proses pelatihan data yang dijalankan diatas *cloud server virtual machine* dengan GPU dan RAM yang mumpuni. Tahap keempat adalah proses analisis

dari hasil yang didapatkan. Kemudian tahap kelima adalah proses pembuatan aplikasi Android menggunakan kerangka kerja Flutter, serta penanaman model yang akan digunakan, sehingga aplikasi android ini bertindak sebagai alat pendeteksi yang *portable* dan mudah digunakan. Langkah pengerjaan dan pembuatan aplikasi Quameaty ditunjukkan oleh *flowchart* pada Gambar 1.

C. Implementasi

Dengan konsep sistem yang sudah didapat, akan direalisasikan ke dalam beberapa proses pendeteksian. Pertama, data daging ayam yang akan diamati akan diolah menggunakan *machine learning* dengan metode CNN dan model InceptionV3. Resolusi gambar daging ayam yang akan diolah ini akan diperkecil menjadi dibawah 300 x 300 piksel dan diberi pengkanalan RGB, kemudian dilakukan augmentasi gambar dengan beberapa parameter seperti *rotate*, *shifting*, *zero centered*, dan lain-lain. Setelah itu data akan dilatih yang kemudian nilai akurasi dan *loss* akan diperhatikan untuk dapat dilakukan proses analisis program kedepannya.



Gambar 1. *Flowchart* pembuatan aplikasi Quameaty

Tabel 1. Parameter yang diharapkan

Parameter	Nilai
Akurasi Validasi	$\geq 0,9$
Loss Validasi	$< 0,4$

D. Pengujian

Pengujian berupa pendeteksian kualitas daging ayam dari gambar sampel baru yang mana dalam proses mendapatkan gambar sampel tersebut tetap memperhatikan protokol kesehatan yang berlaku. Pengujian dilakukan secara *cloud based*, hasil tangkapan data uji akan diunggah ke *cloud* untuk diklasifikasikan oleh model yang sudah dilatih.

Parameter yang diharapkan dari model yang akan dibangun ditunjukkan oleh Tabel 1. Tujuan dari dilakukannya pengujian ini adalah untuk meningkatkan kinerja model untuk dapat beradaptasi dan mengklasifikasikan data uji secara benar.

E. Pengembangan Aplikasi Android

Aplikasi Android akan dikembangkan menggunakan Flutter dengan bahasa pemrograman Dart. Proses pengembangan akan memperhatikan kenyamanan dan kemudahan pengguna dalam mengoperasikan aplikasi, maka dari itu dilakukan proses desain UI/UX terlebih dahulu sebelum menuju tahap pengembangan secara program.

Proses integrasi antara model pelatihan daging ayam dan aplikasi Android akan dilakukan dengan melakukan proses penanaman model TensorFlow yang telah dikonversi menjadi format TFLite menggunakan *package tf lite* pada Flutter.

III. HASIL DAN PEMBAHASAN

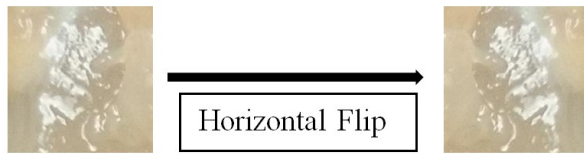
A. Persiapan Data

Data yang telah dikumpulkan akan berupa *video*, yang kemudian diambil beberapa gambar pada setiap *frame per second*. Data akan diberi dua label, yaitu segar dan busuk yang seterusnya akan dipakai sampai pada tahap evaluasi. Data gambar *training* dan *testing* tentunya berbeda dan tidak memiliki kesamaan satu sama lain. Hal ini bertujuan agar nantinya model dapat melakukan pendeteksian pada data *testing* dengan mempelajari dahulu data *training* pada proses pelatihan.

Pada proses ini, didapat sebanyak 28.401 gambar, dengan pembagian antara data latih dan validasi adalah 70 berbanding 30 persen dengan sebaran yang dapat dilihat pada Tabel 2. Data *testing* berjumlah 1536 gambar yang nantinya data ini digunakan dalam proses uji coba atau *testing*.

Tabel 2. Sebaran data

Data	Segar	Busuk
Training	9500	9500
Validation	3890	3975
Testing	1536	
Jumlah	28401	



Gambar 2. Contoh augmentasi dengan memanfaatkan *horizontal flip*

Tabel 3. Parameter augmentasi

Parameter	Nilai
Rotation Range	180°
Horizontal Flip	True
Shear Range	0,2
Rescale	1/255
Width Shift Range	0,2
Height Shift Range	0,2

B. Augmentasi Gambar

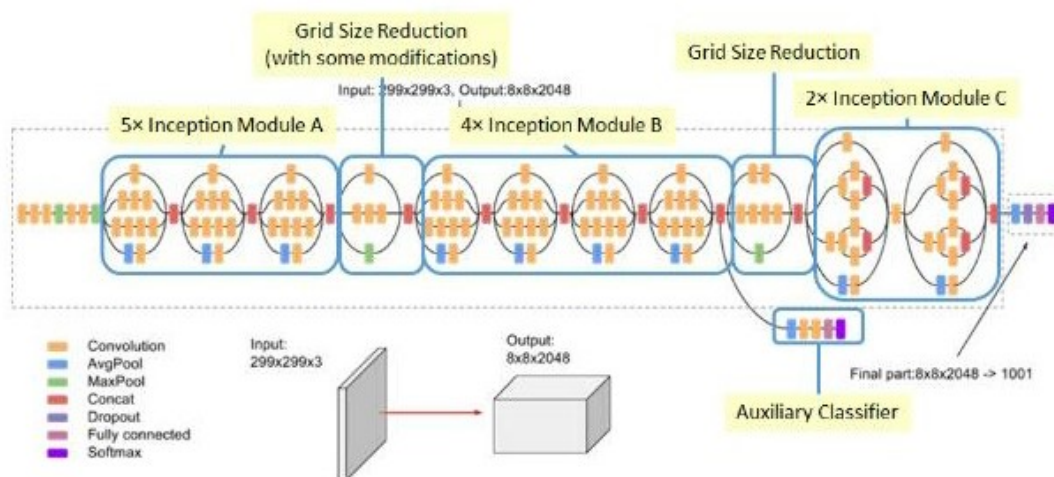
Proses augmentasi gambar dilakukan baik pada data latih maupun validasi dengan parameter yang serupa dengan tujuan untuk memperbanyak dataset dengan cara memodifikasi versi dari data yang sudah ada. Augmentasi ini dilakukan dengan memanfaatkan objek ImageDataGenerator dari pustaka TensorFlow Core v2.6.0. Parameter augmentasi ditentukan tanpa mengubah pengertian daging ayam itu sendiri, sehingga parameter seperti menginversi gambar tidak dilakukan karena pada praktiknya, tidak terdapat daging ayam yang jelas berwarna hitam putih untuk dilakukan pengamatan.

Proses augmentasi gambar bertujuan untuk memperbanyak data latih serta untuk dapat mengenali berbagai perubahan yang mungkin terjadi pada gambar yang disesuaikan dengan kondisi yang sebenarnya di lapangan. Berikut dibawah ini diperlihatkan salah satu data yang di augmentasi menggunakan horizontal flip kemudian diikuti oleh tabel parameter augmentasi yang diterapkan.

C. InceptionV3

InceptionV3 adalah sebuah model yang sangat kompleks sebagai terobosan baru untuk CNN. Beberapa model pendahulunya telah memberikan hasil yang cukup baik berupa nilai akurasi yang tinggi, akan tetapi memiliki masalah pada waktu komputasi yang disebabkan oleh jumlah parameter yang begitu banyak dan tidak menyebar pada berbagai lapisan.

Arsitektur InceptionV3 terdiri dari banyak filter yang disebar pada beberapa lapisan dalam satu waktu pemrosesan yang sama [10]. Pada modul yang pertama yaitu Module A, terdapat lapisan konvolusi kecil berukuran 3×3 yang bertujuan untuk mengekstraksi fitur unik yang masih bersifat umum pada gambar serta melakukan peningkatan performa pada model dengan mengurangi jumlah parameter yang ada. Modul B berguna untuk melakukan pengenalan fitur lebih dalam dengan menggunakan empat lapisan konvolusi masing-masing dua untuk $1 \times n$ dan $n \times 1$. Modul C melakukan penyebaran filter untuk mempertahankan kemungkinan *loss feature* yang ada. Jaringan InceptionV3 memiliki kedalaman sebesar 42 lapisan, dan masih lebih baik serta efisien dibandingkan dengan pendahulunya seperti VGGNet.



Gambar 3. Arsitektur InceptionV3 [10]

```

# Download Inception v3 weights to local machine
!wget --no-check-certificate \
  https://storage.googleapis.com/mledu-datasets/inception_v3_weights_tf_dim_ordering_tf_kernels_notop.h5 \
  -O /tmp/inception_v3_weights_tf_dim_ordering_tf_kernels_notop.h5

# Import Inception v3 Model
from tensorflow.keras.applications.inception_v3 import InceptionV3
pre_trained_model = InceptionV3(input_shape=(150, 150, 3), include_top=False, weights=None)

# Load Inception v3 weights into model
local_weights_file = '/tmp/inception_v3_weights_tf_dim_ordering_tf_kernels_notop.h5'
pre_trained_model.load_weights(local_weights_file)

#set layer to be non trainable
for layer in pre_trained_model.layers:
    layer.trainable = False

#Obtain last layer output of the pre-trained model
last_layer = pre_trained_model.get_layer('mixed7')
last_output = last_layer.output

```

Gambar 4. Program *transfer learning* InceptionV3

Pada dasarnya, model InceptionV3 telah hadir bersama lapisan Fully-Connected atau lapisan Dense. Lapisan ini memiliki peran sebagai penghubung antara input dengan unit aktivasi dengan tujuan untuk melakukan pencocokan fitur yang telah dipelajari pada lapisan konvolusi dan *pooling*.

Proses *transfer learning* yang dilakukan akan menggunakan InceptionV3 sebagai susunan lapisan pertama, kemudian lapisan *Fully-Connected* akan dirancang secara mandiri. Proses *transfer learning* ditunjukkan oleh Gambar 4.

D. Lapisan Model

Sebagai awal, *pre-trained* model InceptionV3 yang diperoleh dari repositori Google merupakan model yang sangat kompleks dan lengkap. Pada pelatihan ini, *layer Fully-Connected* bawaan akan dibuang, dan dibuat kembali dengan jumlah unit filter yang lebih sedikit yaitu 1024. Model InceptionV3 digunakan di awal untuk proses konvolusi, *max pooling*, dan *average pooling*. Susunan lapisan model ditunjukkan oleh Tabel 4.

E. Hasil dan Evaluasi Model

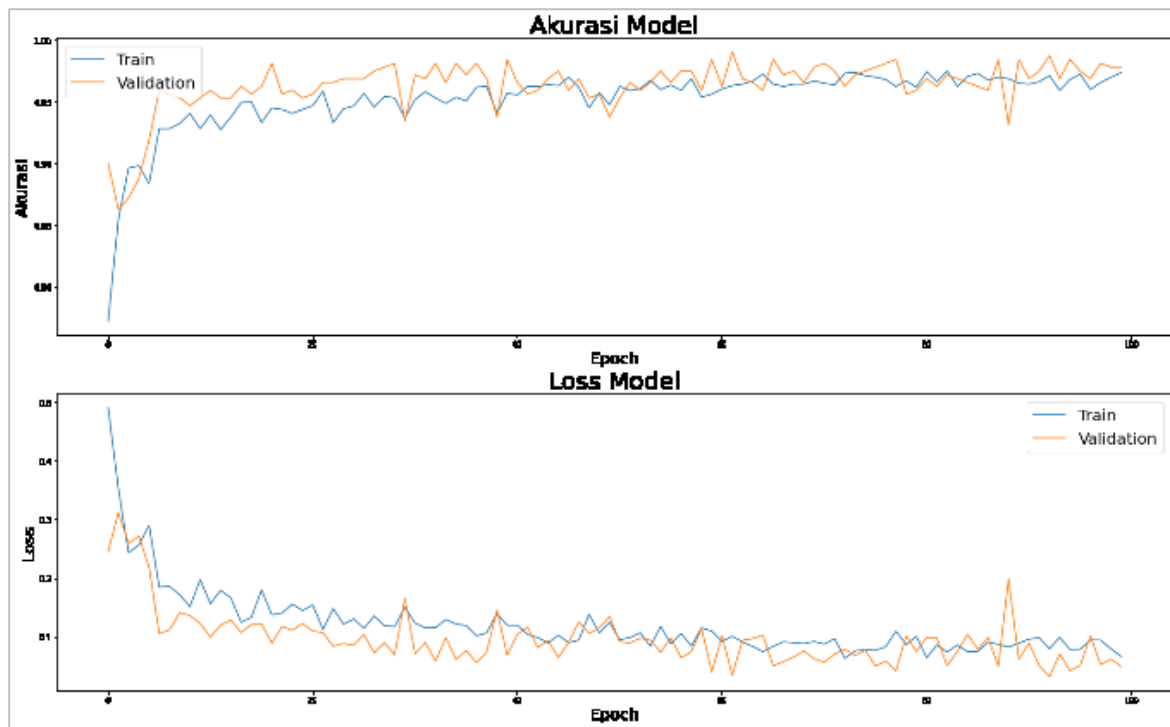
Proses pelatihan dilakukan menggunakan data *training* untuk kedua label yaitu daging segar dan busuk berjumlah 19 ribu gambar. Proses pengujian dilakukan secara otomatis menggunakan bahasa pemrograman Python untuk melakukan proses input gambar hingga pendeteksian. Dari proses pelatihan atau *training* yang telah dilakukan sebanyak 100 *epoch* dengan 50 *steps* di tiap epochnya, didapat nilai rata-rata akurasi dan *loss* yang telah mencapai target awal yaitu untuk *validation accuracy* $\geq 0,9$ dan *validation loss* $< 0,4$ yang dapat diamati dapat dilihat pada Tabel 5.

Tabel 4. Lapisan model

Layer	Keterangan	Parameter / Argumen
<i>Input</i>	Layer untuk menerima gambar <i>input</i> . Dimensi dibuat kecil 150 agar proses pelatihan berjalan efektif.	Size = 150x150x3
InceptionV3	<i>Pre-trained</i> model InceptionV3.	Activation = ReLu
Flatten	Layer untuk meratakan tensor yang telah didapatkan menjadi 1D.	-
<i>Fully Connected</i>	<i>Hidden layer</i> umpan balik untuk melakukan proses prediksi.	Unit = 1024, Activation = ReLu, Dropout = 0,2
<i>Fully Connected</i>	<i>Hidden layer</i> umpan balik terakhir, yang akan membagi hasil prediksi menjadi 2 nilai biner, yaitu 0 untuk prediksi busuk dan 1 untuk segar.	Unit = 2, Activation = softmax

Tabel 5. Hasil metrik pengukuran

Metrik	Nilai
<i>Loss</i>	0,12
<i>Validation Loss</i>	0,096
<i>Accuracy</i>	0,95
<i>Validation Accuracy</i>	0,963



Gambar 5. Grafik metrik terhadap peningkatan *epoch*

Tabel 5. Nilai *confusion matrix* pada *threshold* 50%

		Actual	
		Positive	Negative
Predicted	Positive	TP = 592	FP = 32
	Negative	FN = 26	TN = 886

Tabel 7. Nilai *confusion matrix* pada *threshold* 75%

		Actual	
		Positive	Negative
Predicted	Positive	TP = 580	FP = 17
	Negative	FN = 38	TN = 901

Tabel 6. Hasil metrik pengujian data *testing* pada *threshold* 50%

Metrik	Nilai
Accuracy	0,9622
Precision	0,9487
Recall	0,9579
F1 Score	0,9533

Tabel 8. Hasil metrik pengujian data *testing* pada *threshold* 50%

Metrik	Nilai
Accuracy	0,9641
Precision	0,9715
Recall	0,9385
F1 Score	0,9547

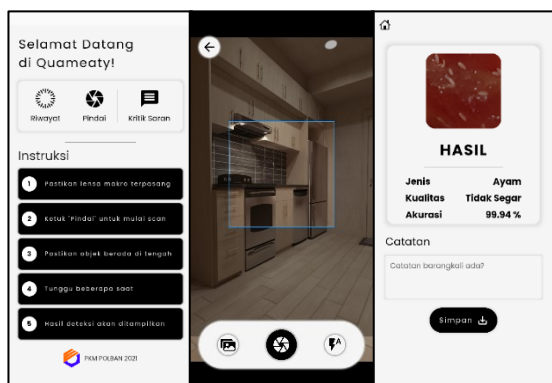
Gambar 5 menunjukkan grafik perubahan nilai akurasi dan *loss* seiring meningkatnya *epoch*. Nilai akurasi cenderung meningkat dan nilai *loss* cenderung menurun, hal ini memperlihatkan bahwa model pelatihan yang dirancang serta pemanfaatan *transfer learning* menggunakan model InceptionV3 telah berjalan dengan baik. Pada 10 *epoch* atau iterasi pertama, terlihat bahwa model masih mencoba mempelajari tiap bagian unik dari gambar, hal ini ditunjukkan oleh grafik yang naik dan turun signifikan pada nilai akurasi dan *loss*.

Selain itu, dihitung pula nilai *confusion matrix* pada *threshold* 50% dan 75% beserta metrik pengukuran lainnya. Proses perhitungan ini dilakukan terpisah dengan program yang berbeda dan menggunakan model yang telah di *export* terlebih dahulu.

Proses perhitungan nilai *confusion matrix* dilakukan secara otomatis menggunakan bahasa pemrograman Python. Pengertian untuk masing-masing nilai *confusion matrix* adalah sebagai berikut:

- TP = *True Positive*, jumlah daging segar yang diprediksi segar.
- FN = *False Negative*, jumlah daging busuk yang diprediksi busuk.
- TN = *True Negative*, jumlah daging segar yang diprediksi busuk.
- FP = *False Positive*, jumlah daging busuk yang diprediksi segar.

Dari kedua pengujian dengan *threshold* yang berbeda, didapat bahwa perbedaan matriks pengukuran pada keduanya tidak jauh berbeda.



Gambar 6. Tampilan aplikasi Android

Akan tetapi pada pengujian dengan *threshold* 75%, nilai *recall* cenderung menurun kurang lebih 0,02, dimana nilai *recall* sendiri ialah nilai sensitifitas yang menunjukkan rasio prediksi benar positif (TP) dibandingkan dengan keseluruhan data yang benar positif dan salah negatif. Proses pelatihan data telah berjalan dengan baik dikarenakan nilai metrik pengujian telah melebihi 90%, sehingga probabilitas prediksi daging segar maupun busuk telah mendekati hasil yang sebenarnya.

F. Pengembangan Aplikasi Android dan Integrasi Model Pelatihan

Aplikasi Android Quameaty telah berhasil dikembangkan dengan baik dan tanpa kendala yang berarti. Digunakan bahasa pemrograman Dart dan SDK Flutter untuk mengembangkan aplikasi Android ini. Tampilan dan fitur aplikasi Android dapat dilihat pada Gambar 6.

Proses integrasi antara aplikasi Android dan model pelatihan TensorFlow yang telah dilatih telah berhasil dilakukan, dimana model perlu dikonversi terlebih dahulu menjadi format TFLite dan digunakan pustaka TFLite untuk Flutter sebagai langkah untuk melakukan penanaman model. Maka dari itu, aplikasi Quameaty ini dapat melakukan pendeteksian kualitas daging ayam mentah.

IV. KESIMPULAN

Dari penelitian yang telah dilakukan yang meliputi proses pelatihan menggunakan model InceptionV3 dan pengujian data menggunakan data *testing*, hasil yang didapatkan cukup baik karena seluruh metrik pengujian telah melebihi nilai 90% baik pada *threshold* 50% maupun 75%. Pada *threshold* 75%, nilai *recall* atau nilai sensitifitas model untuk memprediksi hasil benar positif dibandingkan seluruh nilai benar positif dan salah negatif yang sebenarnya. Proses pelatihan dan uji

coba atau *testing* hanya dilakukan pada dataset yang didefinisikan yang berjumlah 28401 gambar tanpa adanya input gambar diluar dataset tersebut. Proses integrasi aplikasi Android dan model pelatihan telah dilakukan dengan melakukan konversi format TensorFlow ke bentuk TFLite.

UCAPAN TERIMA KASIH

Ucapan terima kasih disampaikan kepada pihak dan bagian Kemahasiswaan Politeknik Negeri Bandung atas bantuan dana yang telah diberikan melalui skema Program Kreativitas Mahasiswa tahun anggaran 2021.

REFERENSI

- [1] J. de Macedo neto, J. dos Santos, and W. R. Schwartz, "Meat adulteration detection through digital image analysis of histological cuts using LBP," Federal University of Minas Gerais, Belo Horizonte, 2016.
- [2] A. Ropodi, F. Mohareb, E. Panagou, and Pavlidis, "Multispectral image analysis approach to detect adulteration of beef and pork in raw meats," *Food Research International*, vol. 67, no. 1, pp. 12-18, 2015.
- [3] M. Al-Sarayreh, M. Reis, and W. Qi Yan, "Detection of Red-Meat Adulteration by Deep Spectral-Spatial Features in Hyperspectral Images," *Journal of Imaging*, vol. 4, no. 5, 2018.
- [4] H. Ayaz, A. Sohaib, M. Mazzara, and M. Ahmad, "Hyperspectral Imaging for Minced Meat Classification Using Nonlinear Deep Features," *Applied Sciences Journal*, vol. 10, no. 1, pp. 1-13, 2020.
- [5] S. Agustin and R. Dijaya, "Beef Image Classification using K-Nearest Neighbor Algorithm for Identification Quality and Freshness," in *Journal of Physics Conference Series*, Tasikmalaya, 2018.
- [6] R. Jorfi, S. Mustafa, Y. Che Man, and D. Mat Hashim, "Differentiation of pork from beef, chicken, mutton and chevon according to their primary amino acids content for halal authentication," *African Journal of Biotechnology*, vol. 11, no. 32, pp. 8160-8166.
- [7] A. Wilson and Viancy, "Meat Monitoring System Using Machine Learning, Internet of Things," *International Journal of Applied Engineering Research*, vol. 14, no. 5, pp. 66-70, 2019.
- [8] E. F. Anggara, T. W. Widodo, and D. Lelono, "Deteksi Daging Sapi Menggunakan Electronic Nose Berbasis Bidirectional Associative Memory," *IJEIS*, vol. 7, no. 2, pp. 209-218, 2017.
- [9] Salsabila, A. Fitrianto, and B. Sartono, "Image Classification Modelling of Beef and Pork Using Convolutional Neural Network," *International*

Journal of Sciences: basic and Applied research (IJSBAR), vol. 57, no. 2, pp. 26-38, 2021.

Computer Vision and Pattern Recognition (CVPR), Las Vegas, 2016.

- [10] C. Szegedy, V. Vanhoucke, and S. Ioffe, "Rethinking the Inception Architecture for Computer Vision," in *IEEE Conference on*