

Implementasi *Convolutional Neural Network* Berbasis Model untuk Klasifikasi Kelayakan Citra Permen Jahe pada Perangkat *Android*

Implementation of Model-based Convolutional Neural Network for Ginger Candy Image Feasibility Classification on Android Device

¹Estiyan Dwipriyoko*), ¹Fabio Syechan Pamungkas, ¹Kusmaya dan ²Dianne Amor Kusuma

¹Universitas Langlangbuana, 40261, Bandung, Indonesia

²Universitas Padjadjaran, 45363, Sumedang, Indonesia

*) *corresponding email: estiyand@gmail.com*

Abstrak

Jahe merupakan salah satu bahan dasar pembuatan permen jahe. Proses evaluasi kelayakan permen jahe di Perusahaan Tasacika yang masih dilakukan secara manual, masih rawan terjadi kesalahan dan kurang efisien. Penelitian ini bertujuan untuk mengembangkan model *Convolutional Neural Network* untuk klasifikasi kelayakan permen jahe dan membuat aplikasi berbasis android yang memudahkan proses tersebut. Metode penelitian menggunakan pendekatan eksperimental. Pengembangan model dilakukan dengan *Convolutional Neural Network* dengan arsitektur MobileNetV2, dan menggunakan metode *Cross Industry Standard Process for Data Mining*. Pengembangan perangkat lunak dilakukan dengan metode *Prototyping*. Penelitian ini menggunakan dataset berupa gambar yang diambil langsung dari pabrik permen jahe milik Perusahaan Tasacika. Model dilatih dan diuji menggunakan *Google Colab* dengan bahasa pemrograman *Python*, serta library *TensorFlow* dan *Keras*. Implementasi dilakukan menggunakan *Kotlin* dan *XML*. Dapat disimpulkan bahwa penelitian ini telah berhasil mengembangkan model klasifikasi kelayakan permen jahe. Hasil pengujian menunjukkan bahwa model yang dikembangkan efektif dalam meminimalisir human error pada proses pengecekan kelayakan permen jahe. Penelitian ini juga berhasil mengembangkan aplikasi klasifikasi kelayakan permen jahe berbasis *Android*.

Kata Kunci: *Permen Jahe, CNN, MobileNetV2, CISP-DM, Prototyping, Android*

Abstract

Ginger is one of the primary ingredients for ginger candy. The manual process of evaluating the feasibility of ginger candy at the Tasacika Company is still prone to errors and is less efficient. This research aims to develop a *Convolutional Neural Network* model for classifying the feasibility of ginger candy and create an *Android*-based application that facilitates this process. The research method uses an *Experimental* approach. Model development is carried out with a *Convolutional Neural Network* with the *MobileNetV2* architecture, using the *Cross Industry Standard Process for Data Mining* methods. Software development is done using the *Prototyping* method. This research used a dataset of images taken directly from the Tasacika Company's ginger candy factory. The model is trained and tested using *Google Colab* with the *Python* programming language and the *TensorFlow* and *Keras* libraries. Implementation is carried out using *Kotlin* and *XML*. It can be concluded that the research has succeeded in developing a ginger candy feasibility classification model. The test results show that the developed model is effective in minimizing human error in the process of checking the feasibility of ginger candy. This research also succeeded in developing an *Android*-based ginger candy feasibility classification application.

Keywords: *Ginger Candy, CNN, MobileNetV2, CISP-DM, Prototyping, Android*

Makalah diterima 16 Maret 2025– makalah direvisi 17 April 2025- disetujui 18 April 2025

Karya ini adalah naskah akses terbuka dengan lisensi [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/).



1 Pendahuluan

Jahe (*Zingiber officinale*) adalah tanaman serbaguna yang banyak dimanfaatkan sebagai rempah, sumber minyak atsiri, pewangi, dan bahan obat. Permen jahe merupakan produk olahan jahe yang dipadukan dengan bahan tambahan lainnya. Seperti produk olahan jahe lainnya, permen jahe memiliki berbagai manfaat

kesehatan, seperti meningkatkan kekebalan tubuh, mengurangi peradangan dan gangguan pencernaan, serta meredakan gejala masuk angin. Produk permen jahe yang dihasilkan dalam program ini memiliki tekstur kenyal seperti jeli, rasa manis, dan sensasi hangat dari jahe [1].

Perusahaan Tasacika merupakan salah satu perusahaan yang bergerak dalam industri pengolahan permen jahe di Kabupaten Sukabumi [2]. Saat ini, proses evaluasi kelayakan permen jahe di Perusahaan Tasacika masih dilakukan secara manual oleh para pekerja. Pada saat pengiriman barang baru ke setiap toko, karyawan melakukan pengecekan sisa permen jahe secara manual dengan cara mengira-ngira untuk menentukan kualitas produk. Proses manual ini seringkali memakan waktu yang cukup lama, rentan terhadap kesalahan manusia, dan kurang efisien serta tidak akurat. Menurut karyawan dan pemilik Perusahaan Tasacika, paparan sinar matahari pada permen jahe dapat mengakibatkan perubahan pada kualitas produk, terutama karena komposisi bahan dasarnya yang meliputi jahe, gula pasir, dan air. Gula yang ditaburi pada permen jahe ketika terkena panas sinar matahari menyebabkan permen tersebut meleleh. Permen jahe yang meleleh ini dinyatakan tidak layak konsumsi. Oleh karena itu, diperlukan solusi yang lebih efisien dan akurat untuk melakukan evaluasi kelayakan permen jahe guna memastikan kualitas produk tetap terjaga [2].

Untuk meningkatkan kualitas kelayakan produk, perlu dilakukan evaluasi terhadap produk yang telah diberikan kepada konsumen sebelumnya, apakah sudah memenuhi harapan dan keinginan konsumen. Studi kelayakan dilakukan untuk mengurangi kesalahan pada suatu produk [2],[3]. Klasifikasi kelayakan permen jahe menjadi langkah penting untuk memastikan kualitas produk yang konsisten. Hal ini disebabkan oleh sensitivitas permen jahe terhadap lingkungan penyimpanan, seperti paparan sinar matahari yang dapat menyebabkan permen jahe meleleh dan mengalami perubahan.

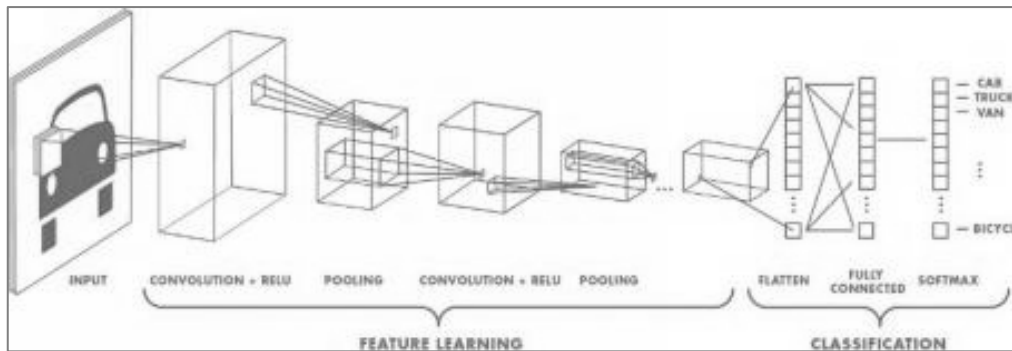
Dalam menangani masalah tersebut, banyak penelitian yang telah dilakukan, termasuk di bidang informatika. Salah satu pendekatan yang sering digunakan adalah teknologi pengolahan citra digital atau *digital image processing*. Metode *Deep Learning* yang populer dalam pengolahan citra digital adalah *Convolutional Neural Network (CNN)*. Ada penelitian yang memanfaatkan metode *CNN* untuk mengklasifikasikan kesegaran buah . Penelitian ini menggunakan *CNN*, untuk menentukan tingkat kematangan buah apel. Hasilnya menunjukkan bahwa model *CNN* dapat mengidentifikasi tingkat kematangan apel dengan akurasi tinggi. Selain itu, aplikasi *Android* yang dikembangkan dari penelitian ini memudahkan pengguna, sehingga meningkatkan efisiensi dalam manajemen panen dan pengolahan apel di sektor pertanian [4],[5],[6].

Berdasarkan latar belakang yang telah dijelaskan, masalah yang dapat dirumuskan adalah sebagai berikut: Apakah model *Deep Learning* yang digunakan dapat mengklasifikasi kelayakan permen jahe dengan memanfaatkan data gambar permen jahe? Apakah penerapan model *Deep Learning* dalam aplikasi dapat mempermudah Perusahaan Tasacika dalam melakukan klasifikasi permen jahe?

Batasan-batasan penelitian agar pembahasan dan isi penelitian ini tetap fokus dan tidak menyimpang dari judul. Batasan-batasan tersebut adalah sebagai berikut: Penelitian ini tidak mencakup proses produksi permen jahe. Aplikasi ini ditujukan untuk karyawan termasuk pemilik Perusahaan Tasacika. Pemodelan *CNN* dilakukan secara diskrit (terputus) dan langsung implementasi. Pemodelan diskrit ini dilakukan karena klasifikasi hanya untuk 2 kondisi. Aplikasi yang dikembangkan tidak menggunakan database untuk menyimpan gambar yang akan mengolah model *CNN* lebih lanjut.

Tujuan dari penelitian ini yaitu sebagai berikut: Mengembangkan model klasifikasi kelayakan permen jahe yang cukup akurat dengan menggunakan *Convolutional Neural Network* untuk meminimalisir kesalahan manusia dalam proses pengecekan. Mengembangkan aplikasi klasifikasi kelayakan permen jahe yang terintegrasi dengan model *CNN* berbasis *Android* untuk memudahkan dan mempercepat pekerjaan Perusahaan Tasacika melakukan dalam proses pengecekan permen jahe.

Convolutional Neural Network atau *CNN* adalah metode jaringan saraf yang populer untuk mengolah data dengan struktur grid, seperti citra dua dimensi dan data berdimensi tinggi seperti video. *CNN* bekerja menggunakan *kernel* dua dimensi atau dimensi tinggi untuk melakukan konvolusi. *Kernel* ini menggabungkan fitur spasial yang menyerupai media input. *CNN* terdiri dari beberapa lapisan *feature learning* dan *classification* [7]. Pada Gambar 1 terdapat *feature learning* yang berguna untuk menerjemahkan suatu input menjadi fungsi berdasarkan sifat-sifat input berupa bilangan dalam vektor. Lapisan ekstraksi ciri ini terdiri dari lapisan konvolusi dan lapisan *pooling*. *Feature* klasifikasi berguna untuk mengklasifikasikan setiap *neuron* dari fitur yang sebelumnya telah diekstraksi [4],[8],[9].



Gambar 1. Arsitektur *Convolutional Neural Network* [9]

Konvolusi menghasilkan transformasi *linear* dari *input* sesuai informasi spasial pada data *input*. Bobot lapisan tersebut melakukan spesifikasi *Kernel* yang digunakan, sehingga *Kernel* dapat dilatih berdasarkan *input*. Kemudian dimensi *output* dari *feature maps* dapat dihitung menggunakan Persamaan (1):

$$Output = \frac{W - N - 2P}{S} + 1 \quad (1)$$

W menunjukkan ukuran *input*, N menunjukkan ukuran *filter (kernel)*, P menunjukkan *zero padding*, S menunjukkan *stride*.

TensorFlow adalah *framework deep learning* dan *library data science open source* yang dikembangkan oleh tim Google. *TensorFlow* dapat digunakan dalam berbagai bidang. Dalam bidang *object detection* terdapat *framework TensorFlow object detection API* yang merupakan suatu alat yang dapat digunakan untuk mempermudah proses *constructing*, *training* dan *deployment* pada suatu model *object detection*. *Framework TensorFlow object detection API* menyediakan *pretrained object detection model* bagi user, namun memungkinkan jika user ingin menggunakan *pretrained object detection model* yang lain, seperti *Faster R-CNN*, *SSD*, *Retinanet*, *Resnet50* dan masih banyak lagi [40],[41].

TensorFlow Lite adalah ekstensi dari *library TensorFlow* yang digunakan untuk transformasi model dan pengoptimalan untuk digunakan pada perangkat seluler. Proses pelatihan terus berjalan di komputer Anda. Setelah proses pelatihan selesai, gunakan *TensorFlow Lite* untuk mengubah dan mengoptimalkan model. Model yang dihasilkan lebih kecil dan lebih optimal, sehingga dapat digunakan pada perangkat *mobile* dengan interpreter *TFLite* [8],[12].

Keras merupakan perangkat lunak jaringan yang berbasis *open source*, dan ditulis dengan menggunakan bahasa pemrograman *Python*. *Keras* juga dapat dijalankan menggunakan *MXNet*, *Tensorflow*, *Deeplearning4j*, dan *Theano* yang khusus dirancang guna mempercepat eksperimen yang berhubungan dengan *Deep Learning*. Fitur yang menonjol dari *Keras* adalah *TensorFlow* dan *Theano* digunakan sebagai *backend*. *Keras* juga dapat berjalan lancar di kedua *CPU* dan *GPU*. *Keras* mendukung hampir semua model jaringan saraf mulai dari *convolutional*, *pooling*, *recurrent*, *embedding*, dan lainnya. Selanjutnya, model ini dapat dikombinasikan untuk membangun model yang lebih kompleks [12],[13].

Cross Industry Standard Process for Data Mining (CRIPS-DM) merupakan alur dari metodologi data mining yang terbagi menjadi enam tahapan yaitu: *business understanding*, *data understanding*, *preparation data*, *modeling*, *evaluation* dan *deployment* [14].

Prototyping adalah proses yang digunakan untuk membantu pengembangan perangkat lunak dengan membentuk model awal perangkat lunak [15]. *Prototype* adalah versi awal dari sebuah tahapan sistem perangkat lunak yang digunakan untuk mempresentasikan gambaran dari ide, mengeksperimenkan sebuah rancangan, mencari masalah yang ada sebanyak mungkin serta mencari solusi terhadap penyelesaian masalah tersebut. Model *prototype* yang dipergunakan oleh sistem akan mengizinkan pengguna mengetahui seperti apa tahapan sistem yang dibuat sehingga sistem dapat mampu beroperasi secara baik. Metode ini dimaksudkan untuk mendapatkan representasi dari pemodelan aplikasi yang akan dibuat, dimulai dari *mockup* yang dievaluasi oleh pengguna, lalu dijadikan rujukan bagi pengembang untuk membangun aplikasi [15].

Kotlin adalah *framework* berbasis *Java Virtual Machine (JVM)* yang berfungsi sebagai bahasa pemrograman pragmatis untuk *Android* yang menggabungkan bahasa fungsional dan berorientasi objek. Karena

interoperabilitasnya, bahasa ini dapat digabungkan dengan *Java* dalam satu proyek dan digunakan untuk mengembangkan aplikasi berbasis *desktop*. *JetBrains*, perusahaan yang mengembangkan *IntelliJ IDEA*, awalnya mengembangkan *Kotlin* dan setelah melakukan banyak perubahan, merilisnya secara *open source*. Saat ini, pengembangannya terus berkembang. *Kotlin* diberikan dukungan penuh oleh *Google* kepada pengembang aplikasi *Android* [15],[16],[17].

Java adalah suatu bahasa pemrograman yang dapat dijalankan di berbagai komputer termasuk telepon genggam. Bahasa pemrograman ini awalnya dibuat oleh *James Gosling* saat itu masih bergabung di *Sun Microsystems*, saat ini merupakan bagian dari *Oracle*, dan dirilis pada tahun 1995. Bahasa ini banyak mengadopsi sintaksis yang terdapat pada bahasa pemrograman *C* dan *C++* namun lebih disederhanakan. Aplikasi-aplikasi berbasis *Java* umumnya dikompilasi ke dalam *p-code* (*bytecode*) dan dapat dijalankan pada berbagai mesin virtual *Java* (*JVM*) [18].

XML singkatan dari *eXtensible Markup Language*, merupakan bahasa markup yang memiliki nilai lebih dibandingkan *HTML*. *XML* merupakan penyederhanaan dari *SGML* (*Standard Generalized Markup Language*) dan direkomendasikan oleh *W3C* pada 10 Februari 1998. *XML* bukan merupakan pengganti *HTML*, namun merupakan pelengkap *HTML*. Masing-masing dikembangkan untuk tujuan yang berbeda. *HTML* digunakan untuk menampilkan informasi dan berfokus pada bagaimana informasi terlihat, *XML* mendeskripsikan susunan informasi dan berfokus pada informasi itu sendiri [19].

Python adalah bahasa pemrograman tingkat tinggi dengan kemampuan untuk mengeksekusi sejumlah instruksi multi guna secara langsung (interpretatif) menggunakan metode *Object Oriented Programming* (*OOP*), meskipun *Python* merupakan bahasa pemrograman tingkat tinggi, tetapi *python* dapat dengan mudah dipelajari karena sudah dilengkapi dengan manajemen memori otomatis [20],[21].

Ada beberapa artikel yang mirip dengan penelitian ini. Klasifikasi kualitas daging marmer berdasarkan citra warna daging menggunakan metode *Convolutional Neural Network* [7], persamaan yang ditemukan adalah penelitian mengimplementasikan metode *Convolutional Neural Network* untuk melakukan klasifikasi berdasarkan citra. Perbedaannya adalah penelitian tentang daging marmer ukuran sedang, sedangkan peneliti berfokus pada kelayakan permen jahe ukuran kecil. Klasifikasi Dan Pengenalan Pola Penyakit Cabai Dengan Metode *CNN* [21], persamaan yang ditemukan adalah penelitian itu memilih *Convolutional Neural Network* sebagai metode untuk melakukan klasifikasi berbasis citra. Perbedaannya adalah penelitian dalam artikel jurnal tersebut tentang identifikasi penyakit pada cabai sedangkan peneliti pada klasifikasi kelayakan permen jahe.

2 Metode

Dalam melakukan penelitian ini, metode yang digunakan adalah metode penelitian Kuantitatif Eksperimental, metode pengembangan model data *CRISP-DM* [14] dan metode pengembangan perangkat lunak *Prototyping* [15]. Metode ini dipilih untuk mengembangkan dan mengevaluasi sebuah aplikasi berbasis *Android* yang mampu mengklasifikasikan kelayakan permen jahe menggunakan model *Convolutional Neural Network* (*CNN*) [9],[21].

2.1 Metode Penelitian Kuantitatif Eksperimental

Metode Penelitian Kuantitatif dengan pendekatan Eksperimental dipilih dalam penelitian ini, untuk membandingkan dua kondisi, yaitu kondisi yang berjalan, dan kondisi usulan dengan manipulasi, yang dibuat dalam penelitian ini [22]. Metode ini mencakup beberapa tahap yang dipaparkan pada Tabel 1.

Tabel 1. Metode Penelitian Kualitatif Eksperimental

Tahapan	Deskripsi
Observasi	Peneliti melakukan pengamatan langsung terhadap proses evaluasi kelayakan permen jahe di PD Tasacika. Ini bertujuan untuk memahami langkah-langkah yang diambil oleh para pekerja dalam mengevaluasi kualitas permen jahe, mulai dari penerimaan barang hingga penentuan kelayakan produk. Observasi ini bertujuan untuk memperoleh pemahaman yang mendalam tentang proses yang sedang berlangsung.
Pengumpulan Data	Setelah observasi dilakukan, peneliti mengumpulkan data berupa citra permen jahe yang akan digunakan untuk pelatihan dan pengujian model <i>Convolutional Neural Network (CNN)</i> . Data ini mencakup gambar permen jahe “layak” dan “tidak layak”, yang dikumpulkan langsung dari pabrik PD Tasacika. Data yang terkumpul ini kemudian akan menjadi dasar untuk membangun model klasifikasi.
Analisis Model	Pada tahap ini, peneliti merancang, melatih, dan mengembangkan model <i>CNN</i> menggunakan data yang telah dikumpulkan. Proses ini mencakup pemilihan arsitektur yang tepat, serta penyesuaian fungsi aktivasi dan parameter lainnya untuk memastikan model berfungsi dengan baik. Model kemudian dievaluasi berdasarkan metrik seperti akurasi untuk memastikan bahwa model dapat mengklasifikasikan permen jahe dengan tepat.
Perancangan Aplikasi (bagian dari siklus <i>Prototyping</i>)	Setelah model <i>CNN</i> dikembangkan, peneliti merancang antarmuka pengguna (<i>User Interface</i>) dari aplikasi <i>Android</i> yang akan mengimplementasikan model <i>CNN</i> tersebut. Untuk mendokumentasikan dan memvisualisasikan desain sistem, peneliti menggunakan <i>Unified Modeling Language (UML)</i> . <i>UML</i> digunakan untuk memberikan gambaran yang jelas dan terstruktur mengenai interaksi antara komponen aplikasi dan alur kerja keseluruhan.
<i>Deployment</i> (bagian dari siklus <i>Prototyping</i>)	Peneliti melakukan pengujian terhadap aplikasi yang telah dikembangkan menggunakan metode <i>Black Box Testing</i> . Tujuan dari pengujian ini adalah untuk memastikan bahwa semua fitur aplikasi berfungsi sebagaimana mestinya dan untuk mengidentifikasi serta memperbaiki bug atau masalah performa sebelum aplikasi diimplementasikan.
Pengujian akhir	Pada tahap akhir, peneliti mengimplementasikan aplikasi dan model <i>CNN</i> dalam bentuk aplikasi berbasis <i>Android</i> . Kemudian, aplikasi diuji dalam lingkungan operasional untuk memastikan bahwa aplikasi siap digunakan oleh pengguna akhir, seperti karyawan atau pemilik PD Tasacika, untuk mengevaluasi kelayakan permen.

2.2 Alat dan Bahan Penelitian

Peneliti menggunakan berbagai perangkat keras dan perangkat lunak untuk mendukung proses penelitian. Berikut adalah daftar alat dan bahan yang digunakan dalam penelitian ini:

2.2.1 Perangkat keras (*hardware*)

Perangkat keras yang digunakan pada penelitian ini seperti yang ditampilkan pada Tabel 2a.

Tabel 2a. Perangkat keras penelitian

Nama alat	Spesifikasi	Fungsi	Pengguna
<i>Laptop</i>	RAM : 16GB SSD : 512GB CPU : AMD Ryzen 5 4600H OS : Windows	Sebagai alat utama dalam membuat model dan aplikasi.	Peneliti
Handphone <i>Android</i>	RAM : 8GB OS : <i>Android 7</i> atau lebih CPU : <i>Quad-core 1.4 GHz</i> atau lebih	Sebagai alat tambahan untuk mencoba dan menguji sistem	Peneliti, Pengguna Akhir

Perangkat keras yang digunakan pada penelitian ini seperti yang ditampilkan pada Tabel 2b.

Tabel 2b. Spesifikasi kebutuhan perangkat keras user *Android*

Kebutuhan hardware	Spesifikasi minimum	Spesifikasi rekomendasi
Sistem operasi	<i>Android 7.0 (Nougat)</i>	<i>Android 10</i> atau lebih
Prosesor	<i>Quad-core 1.4 GHz</i>	<i>Octa-core 2.0 GHz</i>
RAM	2 GB	4 GB atau lebih
Penyimpanan	16 GB	32 GB atau lebih
Kamera	Kamera belakang 8 MP	Kamera belakang 12 MP atau lebih
Layar	Resolusi 720p	Resolusi Full HD (1080p) atau lebih

2.2.2 Perangkat lunak (*software*)

Perangkat lunak yang digunakan pada penelitian ini seperti yang ditampilkan pada Tabel 3.

Tabel 3. Perangkat lunak penelitian

Nama alat	Fungsi	Pengguna
<i>Google Colab</i>	Sebagai <i>software</i> yang berfungsi untuk menguji dan melakukan <i>modelling</i> terhadap <i>dataset</i> .	Peneliti
<i>Android Studio</i>	Sebagai <i>software</i> yang berfungsi untuk membuat aplikasi <i>Android</i> dengan menggunakan <i>framework Kotlin</i> .	Peneliti

2.3 Metode Pengembangan Sistem

Metode pengembangan sistem yang dipakai adalah metode pengembangan perangkat lunak *Prototyping*. Metode *Prototyping* ini terbagi atas 3 iterasi yaitu:

1. Tahap *Prototyping* Model *CNN* menggunakan *TensorFlow* dan *Keras*.
2. Tahap *Prototyping* Model *CNN* konversi ke *TensorFlow Lite*.
3. Tahap *Prototyping* Model *CNN* ke Aplikasi *mobile*.

2.3.1 Tahap *Prototyping* Model *CNN* menggunakan *TensorFlow* dan *Keras*

Peneliti mengumpulkan data dan menganalisis karakteristik dari permen jahe. Dengan data gambar yang berjumlah 2880, terdiri dari dua jenis dengan jumlah yang sama, yaitu 1440 gambar untuk setiap jenis. Berdasarkan data yang peneliti ambil, jenis-jenis permen jahe yang digunakan dalam penelitian ini dijelaskan pada Tabel 4.

Dataset yang terdiri dari total 2880 gambar dibagi menjadi tiga bagian:

1. *Training Set*: Digunakan untuk melatih model. Terdiri dari 80% dari total dataset, yaitu 2304 gambar.
2. *Validation Set*: Digunakan untuk memvalidasi model selama pelatihan guna mengoptimalkan performa. Terdiri dari 10% dari total dataset, yaitu 288 gambar.
3. *Test Set*: Digunakan untuk menguji performa model yang telah dilatih. Terdiri dari 10% dari total dataset, yaitu 288 gambar.

Gambar *dataset* diambil dengan spesifikasi minimal gadget dengan kamera belakang 8 MP, dengan Resolusi 720p. Pengguna direkomendasikan untuk menggunakan spesifikasi gadget dengan kamera belakang 12 MP atau lebih, dengan resolusi *Full HD* (1080p) atau lebih.

Augmentasi citra sangat penting dalam semua masalah *machine learning*, oleh karena itu diperlukan augmentasi terhadap gambar di dataset agar tidak *overfitting* pada *training set*. Salah satu cara augmentasi gambar yang tersedia adalah dengan menggunakan *TensorFlow* dengan bantuan *library package Keras*.

Tabel 4. Hasil *dataset*

Status	Gambar	Ciri-ciri	Jumlah data
Layak		Gula yang ditaburi pada permen jahe terlihat Memiliki warna kuning terang keemasan dan teksturnya kenyal dan kering.	1440
Tidak Layak		Gula yang ditaburi pada permen tidak terlihat dikarenakan sudah meleleh Perubahan warna menjadi coklat tua gelap dan teksturnya yang lengket / basah.	1440

Model dibangun menggunakan arsitektur *MobileNetV2* sebagai *base model*. *MobileNetV2* dipilih karena kemampuannya dalam menghasilkan representasi fitur yang kuat dari gambar dengan *overhead* komputasi yang rendah, menjadikannya ideal untuk aplikasi pada perangkat *mobile*.

```
[ ] base_model = MobileNetV2(include_top=False, weights="imagenet", input_tensor=Input(shape=(224, 224, 3)))
base_model.trainable = False
model = Sequential([
    base_model,
    Conv2D(filters=32, kernel_size=3, activation='relu', padding='same'),
    MaxPooling2D(pool_size=2, strides=2),
    Dropout(0.5),
    Flatten(),
    Dense(1, activation='sigmoid') # For binary classification, use sigmoid activation
])

model.compile(optimizer=Adam(learning_rate=0.001), loss='binary_crossentropy', metrics=['accuracy'])
model.summary()
```

Gambar 2. Membangun model.

Berikut adalah deskripsi dari lapisan-lapisan yang digunakan dalam model berdasarkan Gambar 2 diatas:

1. *Base model*

MobileNetV2 digunakan sebagai lapisan dasar (*base model*) tanpa bagian *fully connected layer* di atasnya (*include_top=False*). Ini berarti bahwa model hanya menggunakan lapisan konvolusi dari *MobileNetV2* yang telah dilatih sebelumnya pada dataset *ImageNet*. Parameter *input_tensor=Input(shape=(224, 224, 3))* menunjukkan bahwa model menerima input gambar dengan ukuran 224x224 piksel dan memiliki 3 saluran warna (RGB). Seluruh lapisan pada *base model MobileNetV2* dibekukan (*trainable=False*), sehingga bobotnya tidak akan diperbarui selama pelatihan. Hal ini memungkinkan pemanfaatan fitur-fitur yang telah dipelajari oleh *MobileNetV2* dari dataset *ImageNet*.

2. Convolutional layer

Lapisan konvolusi tambahan (*Conv2D*) dengan 32 filter, ukuran *kernel* 3x3, fungsi aktivasi *ReLU*, dan *padding* 'same' ditambahkan setelah *base model*. Lapisan ini berfungsi untuk mendeteksi pola-pola visual yang penting dalam klasifikasi gambar.

3. MaxPooling layer

Lapisan *MaxPooling* (*MaxPooling2D*) dengan ukuran *pool* 2x2 dan *stride* 2 digunakan untuk mengurangi dimensi spasial dari output lapisan konvolusi, mempertahankan informasi penting, dan mengurangi jumlah parameter model.

4. Dropout layer

Lapisan *Dropout* (*Dropout*) dengan tingkat *dropout* 0.5 ditambahkan untuk mengurangi *overfitting*. Lapisan ini bekerja dengan cara acak mengabaikan 50% unit dari lapisan sebelumnya selama pelatihan, yang membantu model untuk generalisasi lebih baik pada data baru.

5. Flatten Layer

Lapisan *Flatten* mengubah *output* 2D dari lapisan sebelumnya menjadi vektor 1D, yang memungkinkan input ke lapisan *Dense* berikutnya.

6. Dense Layer

Lapisan *Dense* dengan 1 unit dan fungsi aktivasi *sigmoid* digunakan sebagai lapisan *output* untuk klasifikasi *biner* (layak atau tidak layak). Fungsi aktivasi *sigmoid* menghasilkan nilai probabilitas antara 0 dan 1.

Model CNN dilatih menggunakan *dataset* yang telah dibagi menjadi *training set*, *validation set*, dan *test set*. Pelatihan dilakukan selama 23 *epoch* dengan menggunakan *optimizer Adam* dan fungsi *loss* 'binary_crossentropy'. Metrik akurasi digunakan untuk memantau kinerja model. Proses pelatihan dilakukan dengan cara berikut:

1. Proses pelatihan

Model dilatih selama 23 *epoch* menggunakan generator gambar untuk menghasilkan *batch data* dan *label*. *Optimizer Adam* dengan *learning rate* 0.001 digunakan untuk memperbarui bobot model berdasarkan *error* yang dihitung. Fungsi *loss* yang digunakan adalah 'binary_crossentropy', cocok untuk tugas klasifikasi *biner*. Metrik akurasi digunakan untuk memantau kinerja model selama pelatihan.

2. Kinerja model

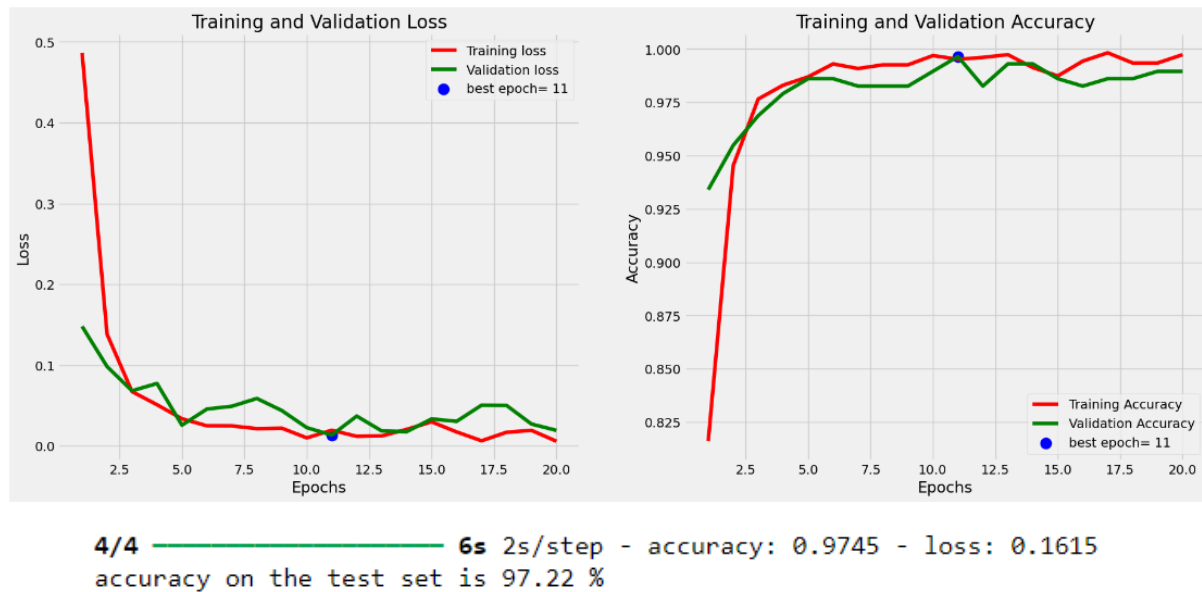
Model dievaluasi menggunakan *validation set* untuk memantau performanya selama pelatihan. Grafik *loss* dan akurasi untuk *training set* dan *validation set* dianalisis untuk memastikan tidak terjadi *overfitting*. Kode yang digunakan untuk melatih model dapat dilihat pada Gambar 3.

```
model.compile(optimizer=Adam(learning_rate=0.001), loss='binary_crossentropy', metrics=['accuracy'])
model.summary()

epochs = 23
history = model.fit(train_gen, epochs=epochs, validation_data=valid_gen)
```

Gambar 3. Pelatihan model.

Setelah pelatihan, grafik *loss* dan akurasi ditampilkan untuk analisis lebih lanjut yang dapat dilihat pada Gambar 4.



Gambar 4. Hasil pelatihan data.

Terlihat pada hasil pelatihan ini bahwa model yang dibangun mendapat akurasi pelatihan mencapai 97% dan loss di angka 0.1615. Keras hanya menghasilkan metrik *Accuracy*.

2.3.2 Tahap *Prototyping* Model *CNN* konversi ke *TensorFlow Lite*

Hasil pelatihan sebelumnya kemudian disimpan ke dalam model *TensorFlow*. Model ini kemudian dikonversi menjadi model *TensorFlow Lite* dengan ekstensi file (.tflite). *TensorFlow Lite* digunakan karena dapat mengurangi ukuran file dan meningkatkan kecepatan eksekusi tanpa memengaruhi keakuratan sehingga dapat digunakan untuk pengembangan aplikasi pada platform *Android*. Berikut adalah kode untuk mengubah ekstensi file model dapat dilihat pada Gambar 5.

```
[ ] model.save('/content/drive/MyDrive/models/cnn_permenjahe.h5')
/usr/local/lib/python3.10/dist-packages/keras/src/engine/training.py:3103: UserWarning: You are saving your model as an HDF5 file via `model.save()`.
saving_api.save_model(

[ ] model_path = '/content/drive/MyDrive/models/cnn_permenjahe.h5'
model = tf.keras.models.load_model(model_path)

[ ] # Convert the model
converter = tf.lite.TFLiteConverter.from_keras_model(model)
tflite_model = converter.convert()

WARNING:absl:'mobilenetv2_1.00_224_input' is not a valid tf.function parameter name. Sanitizing to 'mobilenetv2_1_00_224_input'.
WARNING:absl:'mobilenetv2_1.00_224_input' is not a valid tf.function parameter name. Sanitizing to 'mobilenetv2_1_00_224_input'.
WARNING:absl:'mobilenetv2_1.00_224_input' is not a valid tf.function parameter name. Sanitizing to 'mobilenetv2_1_00_224_input'.

[ ] tflite_model_path = '/content/drive/MyDrive/models/cnn_permenjahe.tflite'
with open(tflite_model_path, 'wb') as f:
    f.write(tflite_model)
```

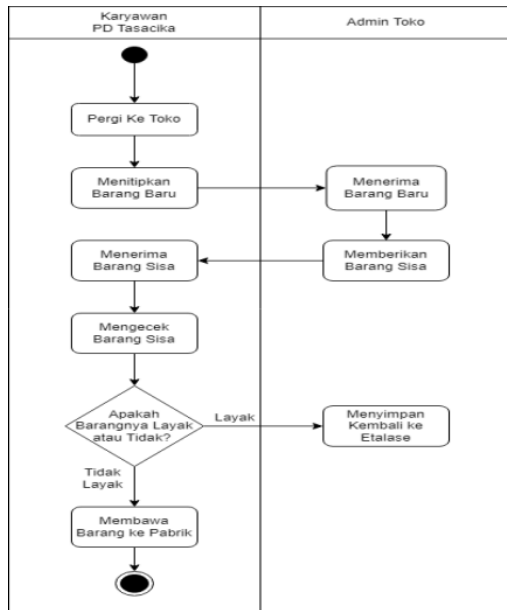
Gambar 5. Mengubah ke *TensorFlow Lite*.

2.3.3 Tahap *Prototyping* model *CNN* ke aplikasi *Mobile*

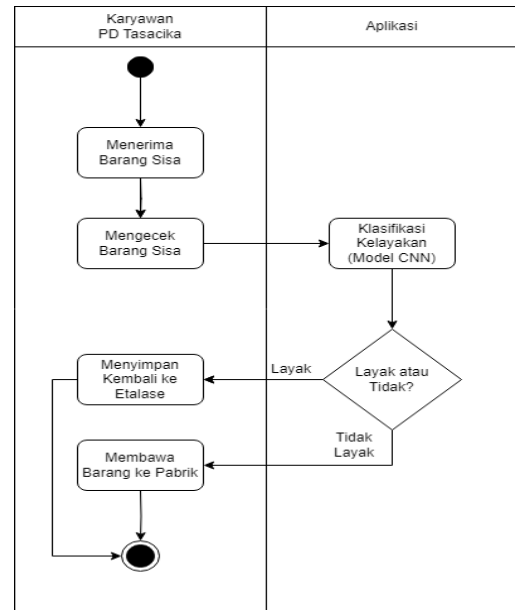
Proses bisnis merupakan rangkaian aktivitas yang dirancang untuk mencapai tujuan tertentu dalam suatu organisasi. Di PD Tasacika, proses ini mencakup seluruh langkah yang diambil oleh karyawan dalam menjalankan tugas sehari-hari, mulai dari pengiriman barang baru, penerimaan barang oleh pegawai toko, hingga pengecekan barang. Proses bisnis berjalan ini ditunjukkan pada Gambar 6.

Pada permasalahan yang dihadapi PD Tasacika, proses bisnis yang diusulkan melibatkan penggunaan aplikasi berbasis *Android* untuk mengklasifikasikan kelayakan permen jahe. Proses bisnis usulan ini bertujuan untuk mempercepat pengecekan kualitas barang, sehingga meminimalkan kesalahan manusia dan meningkatkan kecepatan pengecekan. Proses bisnis usulan ini menggambarkan bagaimana teknologi dapat diintegrasikan

ke dalam alur kerja saat ini untuk mencapai hasil yang lebih baik. Proses bisnis berjalan ini ditunjukkan pada Gambar 7.

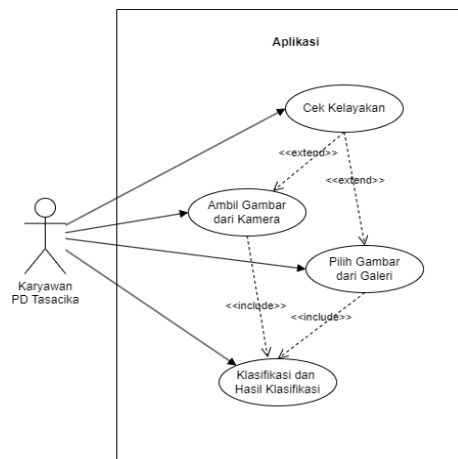


Gambar 6. Proses bisnis berjalan.

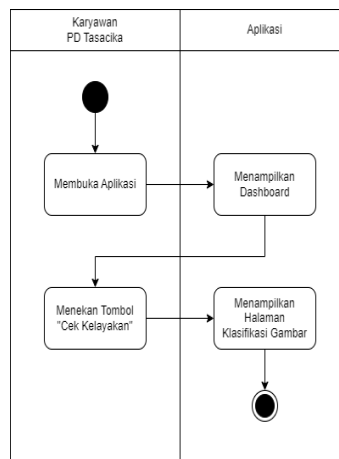


Gambar 7. Proses bisnis usulan.

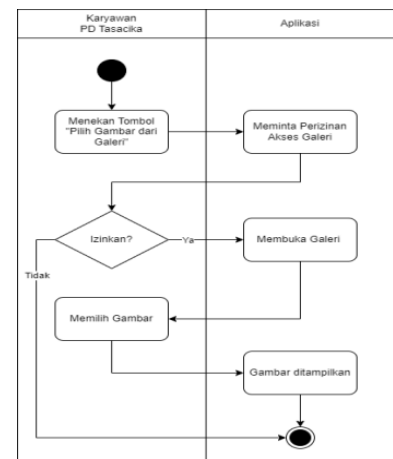
Diagram *Use Case* adalah diagram yang mendefinisikan hubungan antara pengguna dan sistem. Dalam kasus penggunaan untuk aplikasi ini, aktor akan membuka aplikasi, membuka kamera, memilih gambar, lalu mengunggahnya. Selama proses ini, gambar akan melalui proses klasifikasi untuk menentukan kelayakan permen jahe. Berikut *Use Case Diagram* ditunjukkan pada Gambar 8.



Gambar 8. Usecase diagram.

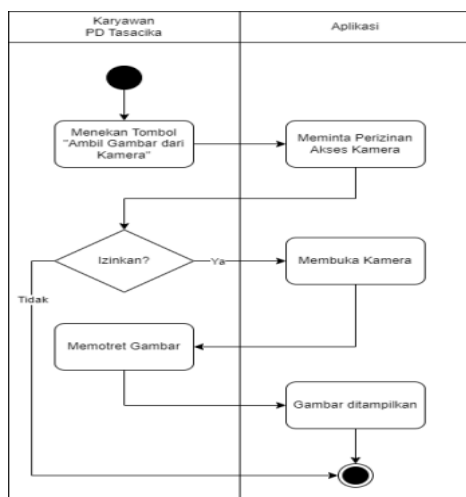


Gambar 9. Activity diagram cek kelayakan.

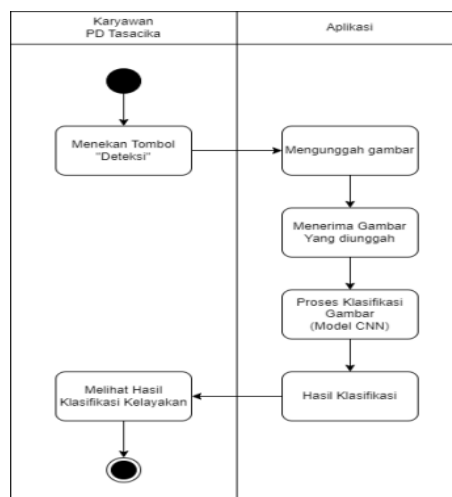


Gambar 10. Activity diagram pilih gambar dari galeri.

Activity diagram adalah grafik yang mendefinisikan alur aktivitas sistem pengguna dan aplikasi. Aktivitas akan dipisahkan oleh sejumlah tabel yang memisahkan pengguna dan sistem. *Activity diagram* ini menunjukkan bagaimana setiap langkah dari pengguna dan sistem berinteraksi dalam proses klasifikasi gambar. Diagram aktivitas aplikasi klasifikasi kelayakan permen jahe dapat dilihat pada Gambar 9, Gambar 10, Gambar 11, dan Gambar 12.

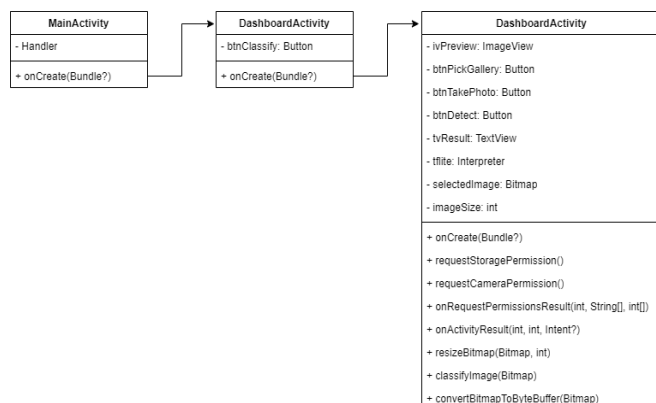


Gambar 11. Activity diagram ambil gambar dari kamera.

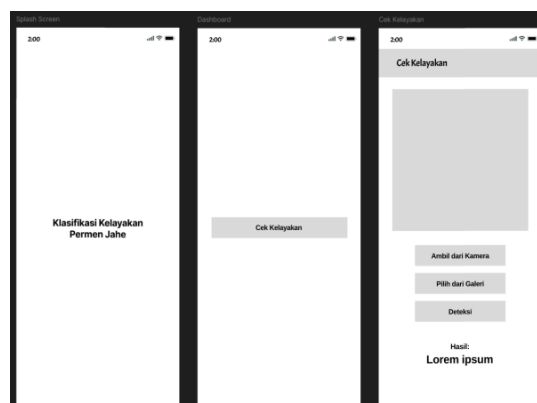


Gambar 12. Activity diagram klasifikasi dan hasil klasifikasi.

Class Diagram menggambarkan susunan sistem dalam aspek pengaturan kelas-kelas yang akan diimplementasikan untuk membangun system. Berikut adalah *Class Diagram* dari aplikasi klasifikasi kelayakan permen jahe dapat dilihat pada Gambar 13.

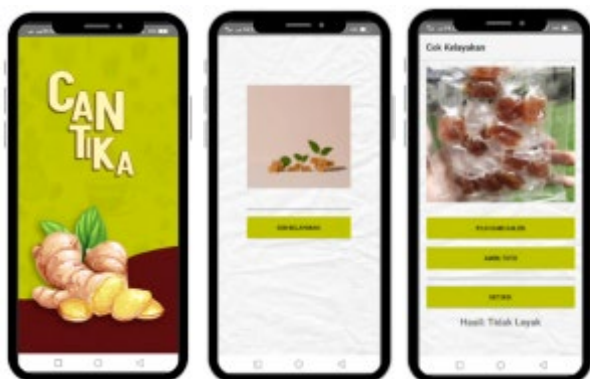


Gambar 13. Class diagram.



Gambar 14. Desain mockup.

Setelah *Mockup* adalah gambaran atau desain visual dari sebuah konsep yang akan diterapkan pada suatu produk. *Mockup* digunakan untuk memberikan pandangan awal tentang bagaimana sistem atau produk akan terlihat dan berfungsi. Desain *Mockup* dari aplikasi klasifikasi kelayakan permen jahe ditunjukkan pada Gambar 14. Desain akhir dari aplikasi yang dibuat ditunjukkan pada Gambar 15.



Gambar 15. Hasil akhir.



Gambar 16. Halaman splash screen.

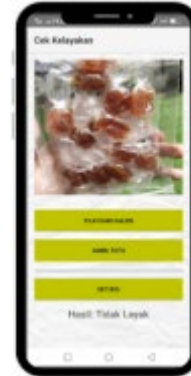
Tampilan awal aplikasi (*Splash Screen*) menampilkan gambar selama 3 detik yang kemudian akan berpindah ke halaman Menu Utama. Tampilan *Splash Screen* ditampilkan dalam Gambar 16.

Tampilan halaman Menu Utama hanya berisi 1 tombol yang menuju ke halaman fitur utama aplikasi ini yaitu klasifikasi kelayakan permen jahe. Halaman Menu Utama ditampilkan pada Gambar 17.

Tampilan halaman Cek Kelayakan merupakan halaman untuk melakukan klasifikasi gambar, yang mana terdapat 3 tombol yaitu tombol “Pilih Dari Galeri” untuk menginputkan gambar dengan cara memilih dari galeri, tombol “Ambil Dari Kamera” untuk menginputkan gambar dengan cara mengambil dari kamera, dan tombol “Deteksi” untuk melakukan proses klasifikasi gambar yang nantinya akan menampilkan hasil klasifikasi. Halaman Cek Kelayakan ditampilkan pada Gambar 18.



Gambar 17. Halaman menu utama.



Gambar 18. Halaman cek kelayakan

3 Hasil

Untuk menghasilkan pengujian klasifikasi yang baik, peneliti melakukan berbagai skenario pengujian guna mencapai hasil yang maksimal. Dalam penelitian ini, pengujian dibagi menjadi dua kategori, yaitu pengujian dengan pencahayaan yang baik atau terang, dan pengujian dengan pencahayaan yang kurang atau gelap.









3.1 Pengujian pencahayaan terang

Pengujian ini dilakukan dengan menggunakan gambar permen jahe yang diambil dalam kondisi pencahayaan yang baik. Pencahayaan yang cukup jelas diperoleh baik dari sinar matahari langsung maupun melalui lampu buatan. Hasil pengujian pencahayaan terang memperlihatkan warna permen jahe, tekstur kelembaban jahe, dan tekstur gula untuk penentuan layak atau tidak. Hasil dari pengujian ini dapat dilihat pada Tabel 5.









3.2 Pengujian pencahayaan gelap

Pengujian ini dilakukan dengan menggunakan gambar permen jahe yang diambil dalam kondisi pencahayaan yang kurang baik. Dalam kondisi ini, detail permukaan permen jahe mungkin kurang terlihat akibat pencahayaan yang tidak memadai. Hasil pengujian pencahayaan gelap memperlihatkan warna permen jahe, dan tekstur gula untuk penentuan layak atau tidak. Hasil dari pengujian ini dapat dilihat pada Tabel 6.

Tabel 5. Cuplikan pengujian pencahayaan terang

No	Sampel gambar	Hasil klasifikasi	Kesimpulan	No	Sampel gambar	Hasil klasifikasi	Kesimpulan
1		Hasil: Layak	Sesuai	5		Hasil: Layak	Sesuai
2		Hasil: Layak	Sesuai	6		Hasil: Layak	Sesuai
3		Hasil: Tidak Layak	Sesuai	7		Hasil: Tidak Layak	Sesuai
4		Hasil: Tidak Layak	Sesuai	8		Hasil: Tidak Layak	Sesuai

Tabel 6. cuplikan pengujian pencahayaan gelap

No	Sampel gambar	Hasil klasifikasi	Kesimpulan	No	Sampel gambar	Hasil klasifikasi	Kesimpulan
1		Hasil: Tidak Layak	Tidak Sesuai	5		Hasil: Layak	Sesuai
2		Hasil: Tidak Layak	Sesuai	6		Hasil: Layak	Sesuai
3		Hasil: Tidak Layak	Sesuai	7		Hasil: Tidak Layak	Sesuai
4		Hasil: Tidak Layak	Sesuai	8		Hasil: Tidak Layak	Tidak Sesuai

3.3 Pembahasan

Hasil pengujian klasifikasi kelayakan permen jahe menunjukkan kinerja yang baik. Model mencapai performa optimal pada *epoch* ke-11 dengan akurasi validasi 97.45% dan *loss* 0.1615. Grafik pelatihan dan validasi menunjukkan penurunan *loss* yang stabil dan peningkatan akurasi yang konsisten, mengindikasikan pembelajaran yang baik tanpa *overfitting* signifikan. Performa model pada test set konsisten dengan akurasi

97.22%, membuktikan kemampuan generalisasi yang baik pada data baru. Namun, perlu diperhatikan bahwa kinerja aplikasi di lapangan dapat dipengaruhi oleh kualitas input gambar, terutama pencahayaan. Untuk memastikan akurasi yang konsisten dalam penggunaan sehari-hari, disarankan untuk mengambil gambar dalam kondisi pencahayaan yang memadai. Penggunaan arsitektur *MobileNetV2* yang dimodifikasi dengan lapisan tambahan berhasil menghasilkan model yang efektif dan ringan untuk klasifikasi kelayakan permen jahe pada aplikasi *Android*. Ini memenuhi tujuan penelitian dalam mengembangkan sistem klasifikasi yang efisien dan andal untuk industri permen jahe, memberikan solusi praktis untuk proses pengecekan permen jahe.

4 Simpulan

4.1 Simpulan

Simpulan dari penelitian ini adalah bahwa aplikasi yang dikembangkan mampu melakukan klasifikasi kelayakan permen jahe menggunakan metode *Deep Learning* yaitu *Convolutional Neural Network (CNN)*. Berikut adalah beberapa simpulan yang dapat peneliti ambil dari penelitian ini, yaitu: Penelitian ini berhasil mengembangkan model klasifikasi kelayakan permen jahe menggunakan metode *Deep Learning* yaitu *Convolutional Neural Network (CNN)*. Hasil pengujian menunjukkan bahwa model yang dikembangkan memiliki akurasi sebesar 97%, sehingga efektif dalam proses pengecekan kelayakan permen jahe. Penelitian ini juga berhasil mengembangkan aplikasi klasifikasi kelayakan permen jahe yang terintegrasi dengan model *CNN* berbasis *Android*, yang memudahkan Perusahaan Tasacika dalam melakukan pengecekan permen jahe. Aplikasi ini memungkinkan pengguna untuk mengambil gambar atau memilih gambar dari galeri dan mendapatkan hasil klasifikasi secara instan, yang meningkatkan efisiensi dan akurasi proses pengecekan. Namun, berdasarkan hasil pengujian, proses klasifikasi di lingkungan dengan cahaya yang kurang atau minim masih menghadapi tantangan dalam melakukan klasifikasi dengan baik, menunjukkan perlunya perbaikan lebih lanjut dalam kondisi pencahayaan yang buruk.

4.2 Saran

Berdasarkan hasil penelitian yang telah peneliti lakukan, saran yang bisa diberikan untuk penelitian selanjutnya adalah: Untuk meningkatkan tingkat akurasi pada aplikasi ini, disarankan untuk mengeksplorasi dan menggunakan metode lain selain *Convolutional Neural Network (CNN)*, serta menggunakan spesifikasi hardware yang lebih tinggi guna mendapatkan hasil yang lebih baik. Penelitian selanjutnya dapat difokuskan pada peningkatan antarmuka pengguna (*User Interface*) dan pengalaman pengguna (*User eXperiences*) dari aplikasi ini untuk meningkatkan kenyamanan dan kemudahan penggunaan aplikasi oleh pengguna. Perlu dilakukan pengujian dan optimasi lebih lanjut terhadap aplikasi di berbagai kondisi pencahayaan untuk memastikan konsistensi dan akurasi klasifikasi permen jahe, termasuk pada kondisi pencahayaan yang kurang atau minim; Penelitian selanjutnya dapat ditambahkan *database* menyimpan gambar untuk mengolah model *CNN* lebih lanjut dan bila ingin membuat klasifikasi yang lebih dari 2 kondisi. *Keras* hanya menghasilkan metrik pengujian *Accuracy*. Untuk mendapatkan metrik pengujian lain seperti *precision*, *recall*, *F1-score* harus menggunakan alat pengujian lain.

Referensi

- [1] M. Melviani, N. Noval & O. Z. Fricillia, "Pembuatan permen jahe sebagai peningkat imunitas tubuh pada masa pandemi." *LOGISTA - Jurnal Ilmiah Pengabdian Kepada Masyarakat*, 6(1), 167 , 2022. <https://doi.org/10.25077/logista.6.1.167-170.2022>
- [2] S. Rohmat, A. Tsani, & N. K. Rini, "Pengaruh segmenting dan targeting terhadap strategi pemasaran permen jahe cantik pada perusahaan Tasacika." *Paspalum: Jurnal Ilmiah Pertanian*, 8(2), 99, 2020. <https://doi.org/10.35138/paspalum.v8i2.192>
- [3] D. Wulandari, & D. Anjani, "Penerapan metode *fuzzy* Tsukamoto untuk mengukur tingkat kelayakan produk." *MIB : Jurnal Media Informatika Budidarma*, 7, 2024–2031, 2024. <https://doi.org/10.30865/mib.v7i4.6879>
- [4] R. Namruddin, Mirfan, & Irfandi, "Klasifikasi kesegaran buah apel menggunakan metode *Convolutional Neural Network (CNN)* berbasis *Android*," *Prosiding SISFOTEK*, 295–302, 2023. Online]. Available: <https://seminar.iaii.or.id/index.php/SISFOTEK/article/download/410/342>
- [5] M. K. Rahmadhika & A. M. Thantawi, "Rancang bangun aplikasi face recognition pada pendekatan CRM menggunakan OpenCV dan algoritma *Haarcascade*," *IKRA-ITH INFORMATIKA: Jurnal Komputer Dan*

- Informatika*, 5(1), 109–118, 2021. [Online]. Available: <https://journals.upi-yai.ac.id/index.php/ikraith-informatika/article/view/921>
- [6] M. S. Nugroho & E. Nurraharjo, “Klasifikasi hama tanaman padi berdasarkan citra daun menggunakan metode *Convolutional Neural Network*.” *BIOEDUSAINS : Jurnal Pendidikan Biologi Dan Sains*, 6(2), 672–682, 2023. <https://doi.org/10.31539/bioedusains.v6i2.8080>
- [7] S. Bagas Valentino, “Klasifikasi kualitas daging marmer berdasarkan citra warna daging menggunakan metode *Convolutional Neural Network*,” *JATI (Jurnal Mahasiswa Teknik Informatika)*, 7(1), 125–129, 2023. <https://doi.org/10.36040/jati.v7i1.6128>
- [8] R. R. M. A. R. Maulana, F. Rizal, & W. J. Shudiq, “Implementasi algoritma *Convolutional Neural Network (CNN)* untuk deteksi kesegaran telur berbasis *Android*,” *Jusikom: Jurnal Sistem Komputer Musirawas*, 8(1), 1–10, 2023. [Online]. Available: <https://jurnal.univbinainsan.ac.id/index.php/jusikom/article/download/1949/1057>
- [9] S. Darmawan Putra Bahari, & U. Latifa, “Klasifikasi buah segar menggunakan teknik computer vision untuk pendeteksian kualitas dan kesegaran buah,” *JATI (Jurnal Mahasiswa Teknik Informatika)*, 7(3), 1567–1573, 2023. <https://doi.org/10.36040/jati.v7i3.6871>
- [10] I. Maryati, “Website perpustakaan ‘Library HUB’ dengan pencarian buku berdasarkan gambar menggunakan Google MLKit,” *JATISI (Jurnal Teknik Informatika Dan Sistem Informasi)*, 8(4), 1821–1831, 2021. <https://doi.org/10.35957/jatisi.v8i4.1269>
- [11] R. Gelar Guntara, “Pemanfaatan Google Colab untuk aplikasi pendeteksian masker wajah menggunakan algoritma *Deep Learning YOLOv7*.” *Jurnal Teknologi Dan Sistem Informasi Bisnis*, 5(1), 55–60, 2023. <https://doi.org/10.47233/jteksis.v5i1.750>
- [12] R. H. Alfikri, M. S. Utomo, H. Februariyanti & E. Nurwahyudi, “Pembangunan Aplikasi Penerjemah Bahasa Isyarat Dengan Metode *CNN* Berbasis *Android*,” *Jurnal Teknoinfo*, 16(2), 183, 2022 <https://doi.org/10.33365/jti.v16i2.1752>
- [13] I. Septian & H. Septanto, “Pengembangan model pendeteksian gambar alat musik dengan metode *Faster R-CNN* dengan *Library Keras*.” *KALBISIANA: Jurnal Mahasiswa Institut Teknologi Dan Bisnis Kalbis*, 8(1), 1–11, 2022. <https://doi.org/10.37012/jtik.v6i2.299>
- [14] Y. Suhandi, I. Kurniati & S. Norma, “Penerapan metode *Crisp-DM* dengan algoritma *K-Means Clustering* untuk segmentasi mahasiswa berdasarkan kualitas akademik.” *Jurnal Teknologi Informatika Dan Komputer*, 6(2), 12–20, 2020. <https://doi.org/10.37012/jtik.v6i2.299>
- [15] E. W. Fridayanthie, H. Haryanto, & T. Tsabitah, “Penerapan metode *prototype* pada perancangan sistem informasi penggajian karyawan (Persis Gawan) berbasis web,” *Paradigma - Jurnal Komputer Dan Informatika*, 23(2), 2021. <https://doi.org/10.31294/p.v23i2.10998>
- [16] J. S. Saputro and U. Latifa, “Prototipe Sistem Peringatan Dini (EWS) bendungan berbasis Internet of Things (IoT) dengan antarmuka web dan aplikasi mobile”, *Jurnal Otomasi Kontrol dan Instrumentasi*, vol. 14, no. 1, pp. 31 - 40, Jun. 2022. <https://doi.org/10.5614/joki.2022.14.1.4>
- [17] A. Febriandirza, “Perancangan aplikasi absensi online dengan menggunakan bahasa pemrograman Kotlin,” *Pseudocode*, 7(2), 123–133, 2020. <https://doi.org/10.33369/pseudocode.7.2.123-133>
- [18] M. D. Payana & H. Pramunsiye, “Perancangan media pembelajaran *English Grammar* berbasis *Android*,” *Journal of Informatics and Computer Science*, 5(2), 110, 2019. <https://doi.org/10.33143/jics.vol5.iss2.548>
- [19] T. Wahyuningrum, “Implementasi *XML Encryption (XML Enc)* menggunakan Java,” *Jurnal Infotel - Informatika Telekomunikasi Elektronik*, 4(1), 17, 2012. <https://doi.org/10.20895/infotel.v4i1.98>
- [20] S. Verma, “Comparative Analysis of Image Classification Algorithms,” *International Journal for Research in Applied Science and Engineering Technology*, 11(12), 1513–1520, 2023. <https://doi.org/10.22214/ijraset.2023.57662>
- [21] B. Nurbuana, T. Cahya, E. Nasrinatun, M. P. Arifin, M. Ayu, & D. Widya, “Klasifikasi dan pengenalan pola penyakit cabai dengan metode *CNN (Convolution Neural Network)*,” *STAINS : Seminar Nasional Teknologi dan Sains*, 3, 125–132, 2024. <https://doi.org/10.29407/stains.v3i1.4137>
- [22] D. Kusuma, E. Dwipriyoko, M. Ihsan & K. Syarif, “Prototipe kendali listrik rumah lewat perintah suara menggunakan jaringan nirkabel dan mikrokontroler,” *Jurnal Tiarsie*, 21(3), 15-24. 2024. <https://doi.org/10.32816/tiarsie.v21i3.252>