BAB I

PENDAHULUAN

A. Latar Belakang

Indonesia, sebagai negara dengan kekayaan budaya yang melimpah, memiliki warisan kuliner yang beragam dan bernilai tinggi, salah satunya adalah kue-kue tradisional. Kue tradisional Bugis, yang berasal dari Sulawesi Selatan, merupakan bagian integral dari warisan kuliner Nusantara yang mencerminkan identitas dan kearifan lokal masyarakat Bugis (Nahriana & Tawani, 2022; Anisa dkk., 2022). Keanekaragaman jenis, bentuk, rasa, dan makna filosofis yang terkandung dalam setiap kue tradisional Bugis menjadikannya aset budaya yang penting untuk dilestarikan (Nahriana & Tawani, 2022; Sudirman dkk., 2021). Namun, seiring dengan derasnya arus globalisasi dan modernisasi, eksistensi kue-kue tradisional ini menghadapi tantangan serius. Terjadi pergeseran minat konsumen dari makanan tradisional ke makanan cepat saji, yang salah satunya disebabkan oleh menurunnya tingkat pengetahuan tentang makanan tradisional dari generasi ke generasi (Sudirman dkk., 2021). Akibatnya, pengetahuan dan minat generasi muda terhadap kue tradisional cenderung menurun, yang berpotensi pada hilangnya warisan budaya ini (Andriyanti & Setiawan, 2024). Generasi muda, khususnya, semakin kurang mengenalnya, yang menyebabkan penurunan minat dan menimbulkan risiko bagi pelestarian warisan budaya yang tak ternilai ini.

Kurangnya dokumentasi yang komprehensif dan media pengenalan yang menarik menjadi salah satu isu utama dalam upaya pelestarian kue tradisional

Bugis. Masyarakat, khususnya generasi muda, seringkali kesulitan untuk mengenali dan membedakan berbagai jenis kue tradisional Bugis, terutama karena banyaknya variasi dan terkadang kemiripan tampilan antar jenis kue (Sumarlie dkk., 2022). Keterbatasan akses terhadap informasi mengenai sejarah, bahan, cara pembuatan, hingga makna budaya dari kue-kue tersebut juga menjadi kendala. Oleh karena itu, urgensi untuk mengembangkan solusi inovatif guna melestarikan kue tradisional Bugis menjadi sangat penting. Pemanfaatan teknologi digital, khususnya aplikasi berbasis perangkat bergerak (*mobile*), menawarkan potensi besar sebagai media edukasi dan pelestarian yang interaktif dan mudah dijangkau oleh masyarakat luas, mengingat tingginya penetrasi pengguna Android di Indonesia (Yunus & Anwar, 2022). Aplikasi berbasis Android dapat dirancang untuk mempermudah proses pengenalan dan penyebaran informasi mengenai makanan tradisional Bugis, termasuk makna yang terkandung di dalamnya (Sudirman dkk., 2021).

Beberapa penelitian sebelumnya telah berfokus pada inventarisasi dan dokumentasi makanan tradisional, termasuk kue-kue dari berbagai daerah di Indonesia (Nahriana & Tawani, 2022; Anisa dkk., 2022). Sudirman dkk. (2021) telah merancang bangun aplikasi ensiklopedia makanan tradisional Bugis berbasis Android yang menyajikan informasi dan cara pembuatannya. Penelitian lain oleh Hasnawati & Asriadi (2022) mengembangkan aplikasi game memasak kue tradisional Bugis menggunakan Construct 2 berbasis Android sebagai media pembelajaran. Di sisi lain, penelitian terkait pemanfaatan teknologi pengenalan citra untuk klasifikasi telah dilakukan, menunjukkan potensi akurasi yang baik (Sumarlie dkk., 2022; Hestiningsih dkk., 2023). Aplikasi berbasis Android juga

telah dikembangkan untuk berbagai keperluan, termasuk penerjemahan bahasa isyarat menggunakan metode *Convolutional Neural Network* (Yunus & Anwar, 2022) dan klasifikasi objek lainnya (Abdiansyah dkk., 2024), yang menunjukkan kelayakan platform ini untuk diseminasi informasi. Namun, pengembangan aplikasi Android yang secara spesifik dirancang untuk pengenalan kue tradisional Bugis dengan mengintegrasikan fitur identifikasi visual melalui kamera atau galeri secara real-time, katalog informasi, dan akses langsung ke resep setelah deteksi. Penelitian ini berupaya mengisi celah tersebut dengan mengembangkan sebuah aplikasi yang tidak hanya mampu mengenali kue tradisional Bugis melalui gambar, tetapi juga menyediakan informasi resep yang relevan dan memungkinkan pengguna untuk membagikan informasi tersebut. Inovasi ini terletak pada fungsionalitas terpadu yang ditawarkan dalam satu platform mobile yang didedikasikan untuk kue tradisional Bugis, guna meningkatkan apresiasi, mempermudah akses informasi, dan mendorong interaksi pengguna.

Penelitian ini bertujuan untuk merancang dan membangun sebuah "**Aplikasi Pengenalan Kue Tradisional Bugis Menggunakan Metode *Convolutional Neural Network* (CNN) Berbasis Android**". Diharapkan aplikasi ini tidak hanya berfungsi sebagai alat identifikasi yang akurat, tetapi juga sebagai media edukasi yang menarik bagi masyarakat, wisatawan, dan generasi muda untuk lebih mengenal, menghargai, dan pada akhirnya turut serta melestarikan warisan budaya kuliner suku Bugis.

B. Rumusan Masalah

Berkaitan dengan pemaparan latar belakang penulis yang sudah ditelaah. Maka rumusan masalah yang hendak dijawab diungkapkan adalah "Bagaimana merancang aplikasi pengenalan kue tradisional bugis menggunakan metode *Convolutional Neural Network* (CNN) berbasis android".

C. Batasan Masalah

Penelitian ini menjumpai segenap keterbatasan, di antaranya:

1. Peranti yang dipakai guna mendeteksi kue tradisional bugis adalah perangkat mobile berbasis system operasi Android.
2. Fitur deteksi kue tradisional bugis hanya mampu mengenali 5 (lima) jenis kue tradisional Bugis tertentu yaitu kue barongko, kue dange, kue lapis, kue onde-onde, dan kue taripang.
3. Fitur katalog akan menampilkan informasi mengenai 10 jenis kue tradisional Bugis, termasuk yang tidak dapat dikenali oleh fitur deteksi kue.

D. Tujuan

Berdasarkan rumusan masalah di atas, tujuan penulisan penelitian ini adalah untuk mengetahui bagaimana cara membuat aplikasi pengenalan kue tradisional bugis menggunakan metode *Convolutional Neural Network* (CNN) berbasis Android.

E. Manfaat Penelitian

Dari judul penelitian di atas yaitu, "Aplikasi Pengenalan Kue Tradisional Bugis Menggunakan Metode Convolutional Neural Network (CNN) Berbasis

Android. Penulis memiliki harapan sanggup membagikan faedah ala teoritis maupun praktis. Adapun manfaat yang diharapkan peneliti, diantaranya:

1. Manfaat Teoritis
    a) Dalam peran serta dalam memperluas pengetahuan dan wawasan terkait suku bugis.
    b) Dalam kapasitasnya sebagai pustaka acuan kepada penelitian-penelitian berikutnya yang berasosiasi kepada aplikasi pengenalan berbasis Android.
2. Manfaat Praktis
    a) Hasil penelitian ini diinginkan guna memperkenalkan serta memberikan informasi pengetahuan dan wawasan terkait olahan masakan kue tradisional bugis bagi masyarakat.
    b) Bagi pihak peneliti, penelitian ini dapat meningkatkan pengetahuan secara teoritis maupun praktis terkait pemograman Android dan kue tradisional bugis.

F. Sistematika Penulisan

Untuk membuat pembahasan dalam tulisan ini lebih mudah dipahami, perlu dijelaskan sistem yang menyusunnya dan pedoman yang digunakan. Sebagai berikut:

BAB I : PENDAHULUAN

Bab ini menguraikan latar belakang, perumusan masalah, batasan masalah, tujuan penelitian, metodologi penelitian, serta struktur penulisan.

BAB II : TINJAUAN PUSTAKA

Bab ini mengulas penelitian-penelitian terdahulu serta landasan teori yang mendukung dalam penyusunan tugas akhir ini..

BAB III : METODE PENELITIAN

Bab ini menjelaskan lokasi dan waktu penelitian, jenis penelitian yang dipakai, metodologi pengumpulan data, alat dan bahan yang digunakan, tahapan penelitian, metode pengujian, serta analisis sistem yang diterapkan.

BAB IV : HASIL DAN PERANCANGAN SISTEM

Bab ini menjelaskan proses desain sistem, termasuk cara integrasi masing-masing komponen yang digunakan. Selain itu, pengujian dilakukan untuk memverifikasi fungsionalitas masing-masing komponen.

BAB V: PENUTUP

Bab ini menyajikan kesimpulan yang diambil dari hasil penelitian dan saran untuk pengembangan lebih lanjut dalam bidang yang diteliti.

DAFTAR PUSTAKA

LAMPIRAN

BAB II

TINJAUAN PUSTAKA

A. Kajian Teori

1. Suku Bugis

Terdapat 34 Provinsi di Negara Kesatuan Republik Indonesia, salah satunya ialah Sulawesi Selatan. Di dalamnya berdian empat kaum utama, yaitu Suku Makassar, Suku Mandar, Suku Toraja, serta Suku Bugis. Satu dari sekian kelompok etnis yang besar jumlahnya yang menetap dan tersebar di daerah Sulawesi Selatan ialah Suku Bugis. Menurut Kapojos dan Wijaya (2018) Suku Bugis sebagian besar tinggal di Kabupaten-Kabupaten Soppeng, Bulukumba, Wajo, Bone, Barru, serta Kota Pare-pare dan Makassar, disebut dengan orang Bugis-Makassar. Suku Bugis tergolong dalam rumpun Deutero-Melayu. Sebutan "Bugis" berakar dari ungkapan "To Ugi", yang tafsirannya ialah "Orang Bugis".

2. Kue Tradisional

Makanan serta minuman yang lazim disantap telah dan diterima oleh sekalian orang dalam suatu masyarakat tertentu karena cita rasanya yang khas dikenal sebagai Makanan Tradisional. Menurut Shosana dan Santoso (2014) mengatakan bahwa kue tradisional merupakan makanan yang sudah lama berkembang di daerah dan masyarakat Indonesia sebagai bentuk makanan ringan atau camilan atau kudapan yang bukan merupakan bagian dari menu utama, dan merupakan bagian dari tradisi. Sementara itu, Ginting, dkk (2017) mendefinisikan

bahwa kue tradisional sebagai warisan budaya dan dilestarikan dari generasi ke generasi oleh nenek moyang. Proses pembuatan kue tradisional meliputi berbagai metode seperti pengukusan, pemaggangan, atau penggorengan. Di Indonesia, kue tradisional biasanya diklasifikasikan berdasarkan kadar air yang dikandungnya:

a) Kue basah

Kue basah biasanya tidak bertahan lama dan bertekstur lunak. Karena komposisi bahan utamanya, seperti gula pasir, tepung beras, dan santan yang rentan mengalami perubahan kualitas. Pengukusan, penggorengan, atau perebusan ialah proses umum pembuatan kue basah. Di pasar tradisional, mudah dijumpai kue basah Indonesia.

b) Kue kering

Kue kering biasanya memiliki tekstur sedikit lebih keras dan bertahan lebih lama daripada kue basah. Perbedaan ini juga terlihat pada metode pengolahannya, di mana kebanyakan kue kering dipanggang atau dibakar di dalam oven, berbeda dengan kue basah.

3. Kue Tradisional Bugis

Kue tradisional Bugis merupakan bagian integral dari warisan budaya Suku Bugis di Sulawesi Selatan, yang tidak hanya berfungsi sebagai hidangan kuliner tetapi juga kaya akan nilai matematis, filosofis, dan simbolis dalam berbagai upacara adat serta kehidupan sehari-hari (Anisa, dkk, 2022; Nur, dkk, 2022). Keberadaan kue-kue ini menandakan identitas budaya yang diwariskan secara turun-temurun (Syahrial, dkk, 2020).

Kue tradisional Bugis umumnya terbuat dari bahan-bahan alami yang mudah ditemukan di daerah setempat, seperti tepung beras, tepung ketan, pisang, kelapa, dan gula merah (Anisa, dkk, 2022; Syahrial, dkk, 2020). Menurut Nurfaizah dan Jefri (2018), proses pembuatannya cenderung sederhana dan tidak memerlukan peralatan khusus, melainkan menggunakan alat-alat dapur biasa (seperti yang dikutip dalam Anisa, dkk, 2022). Banyak kue tradisional Bugis juga dibungkus menggunakan daun pisang, yang memberikan aroma khas serta aman karena tidak mengandung bahan kimia berbahaya (Anisa, dkk, 2022).

4. Dasar Pemrograman

Pemrograman adalah tindakan menyelesaikan masalah atau menghasilkan *output* tertentu dengan memberikan instruksi kepada komputer. Pada dasarnya, ada dua komponen utama yang mendukung pengembangan program: algoritma dan bahasa pemrograman**.**

a) Algoritma

Menurut Khairunisa, dkk (2011) algoritma adalah rangkaian tahap-tahap yang ditetapkan dengan jelas dan logis untuk memecahkan masalah atau mencapai tujuan atau keluaran (*output)* tertentu. Algoritma memainkan peran penting dalam pengembangan perangkat lunak, pemrosesan data, kecerdasan buatan, dan berbagai aplikasi lainnya.

Menurut Khairunisa, dkk. (2011) menyebutkan bahwa algoritma memiliki berbagai fungsi utama dalam pemrograman, seperti menyelesaikan masalah secara efisien, mengoptimalkan kinerja sistem, mengotomatisasi tugas rutin, membantu

pengambilan keputusan, mengamankan data, dan menganalisis data untuk mengidentifikasi pola atau tren.

b) Bahasa Pemrograman

Menurut Ismah (2017), bahasa pemrograman dapat dipahami sebagai alat untuk merancang serangkaian instruksi guna menuntuskan persoalan memakai komputer. Dengan bahasa pemrograman, programmer dapat menetapkan data olah serta informasi yang diharapkan akan dihasilkan. Bahasa pemrograman terdiri dari rangkaian pernyataan yang saling berhubungan, yang mengeluarkan hasil (*output*) sewaktu dieksekusi dan memperoleh masukan (*input*).

Sejak sebelum tahun 1940-an, bahasa pemrograman telah berkembang pesat. Pada awalnya, bahasa ini digunakan untuk kebutuhan spesifik. Namun, kini telah berkembang menjadi dua jenis utama, yaitu:

1) Bahasa pemrograman khusus, seperti SQL bagi pembuatan basis data serta OpenGL bagi grafik.
2) Bahasa pemrograman serba guna seperti Python, C, C++, Pascal, Basic, Java, dll. Digunakan demi beragam kebutuhan mencakup basis data, aplikasi, permainan, dan grafik.

Menurut Ismah (2017), dalam proses pengembangannya, bahasa pemrograman melibatkan tahap kompilasi untuk memeriksa kebenaran kode dan mengubahnya menjadi bahasa mesin. Satu dari sekian bahasa pemrograman yang terkenal ialah Python, yang dikenal dengan sintaksisnya yang sederhana dan keberagaman perpustakaannya (*library*) yang mendukung berbagai aplikasi, mulai dari pengembangan web hingga analisis data.

5. Android

Menurut Satyaputra dan Aritonang (2016) Android ialah sebuah sistem operasi (*operation system*) bagi tablet dan smartphone. Sistem operasi boleh digambarkan sebuah "jembatan" yang menghubungkan pengguna (*user*) dengan peranti (*device*), sehingga memungkinkan pengguna berinteraksi serta berbagai aplikasi yang tersedia di dalamnya. Sementara itu, Menurut Yudhanto dan Wijayanto (2017) Android sebagai sistem operasi berbasis Linux mencakup tak sekedar sistem operasi itu sendiri, melainkan pula *middleware* serta aplikasi bagi perangkat seluler. Berikutnya Murtiwiyati dan Lauren (2013) menyatakan Android ialah sistem operasi untuk perangkat seluler dan mencakup aplikasi, *middleware*, serta sistem operasi itu sendiri.

Sebagaimana telah diutarakan dalam beberapa definisi di atas, dapat disimpulkan bahwa Android ialah sistem operasi (operation system) dengan basis Linux untuk perangkat genggam. Android bersifat *open source*, yang memungkinkan pembuatan aplikasi bagi para pengembang.

6. Android Studio

Android Studio dikembangkan oleh IntelliJ, yang merupakan *Integrated Development Environment* (IDE) milik Google bagi pemrograman Android. Terdapat beragam fitur demi melancarkan para programmer terkhusus programmer pemula yang ditawarkan lewat Android Studio. Selain memiliki banyak fitur, sejumlah pustaka (*library*) siap pakai juga dilengkapi di dalam Android Studio. Meskipun begitu, Android Studio cenderung mengonsumsi lebih banyak *memory*, tetapi ini sebanding dengan manfaat yang ditawarkannya.

Dalam setiap proyek yang dikembangkan di Android Studio memiliki satu atau lebih modul yang mencakup sumber daya dan kode sumber. Modul-modul tersebut meliputi:

a) Modul pustaka
b) Modul Google App Engine
c) Modul aplikasi Android

Struktur folder setiap modul aplikasi mencakup sebagaimana berikut:

a) Manifests: Memuat berkas AndroidManifest.xml.
b) Java : Terdapat file kode sumber Java yang mencakup kode pengujian Junit.
c) Res : Terkandung beragam sumber daya selain kode, semisal *string* UI (*User interface*)*, layout* XML, dan gambar-gambar grafis seperti ikon serta animasi.

7. *Convolutional Neural Network*

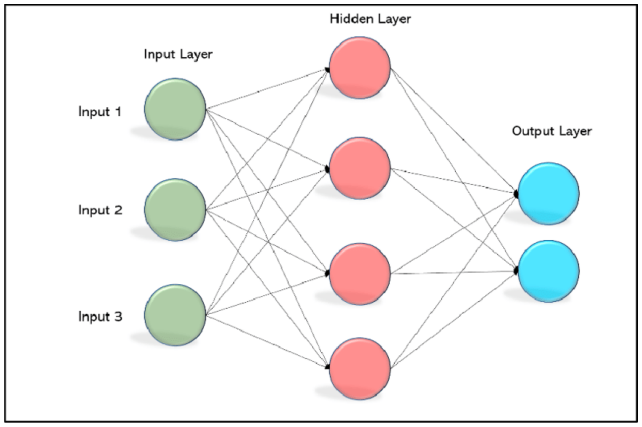
a) Pengertian *Convolutional Neural Network*

Convolutional Neural Network (CNN) merupakan salah satu model deep learning yang unggul karena mampu menggabungkan proses ekstraksi ciri (fitur) dan klasifikasi secara otomatis tanpa perlu dilakukan secara terpisah (Wardani, dkk, 2022). Model ini dirancang secara khusus untuk mengolah data yang memiliki struktur seperti grid, salah satunya adalah citra dua dimensi (I Wayan Suartika, dkk, 2016). CNN memiliki berbagai arsitektur yang telah terbukti efektif, dengan AlexNet menjadi salah satu arsitektur pertama yang sangat dikenal luas untuk tugas klasifikasi citra skala besar (Krizhevsk, dkk, 2012). Selain citra, kemampuan CNN juga dapat diterapkan untuk memproses data dengan dimensi yang lebih tinggi

seperti video, misalnya untuk melakukan klasifikasi aktivitas manusia (Nugraha & Setyati, 2021). Sementara itu, Menurut I Wayan Suartika dkk. (2016), CNN merupakan pengembangan dari Multilayer Perceptron (MLP) yang didesain secara khusus untuk mengolah data dua dimensi dan umum digunakan pada data citra.

Tahun 1980, Kunihiko Fukushima, peneliti asal NHK Broadcasting Science Research Laboratories di Jepang, menggagas NeoCognitron, yang pada masa mendatang dikenali sebagai leluhur dari CNN. Lalu, Yann LeChun dari AT&T Bell Laboratories di Holmdel, New Jersey, USA, menyempurnakan konsep in lebih lanjut. Dalam kajiannya yang tertuju pada pengenalan angka serta tulisan tangan, ia pun berhasil menerapkan suatu model CNN yang dikenal dengan nama LeNet (LeCun, dkk., 1998).

b) Arsitektur *Convolutional Neural Network*



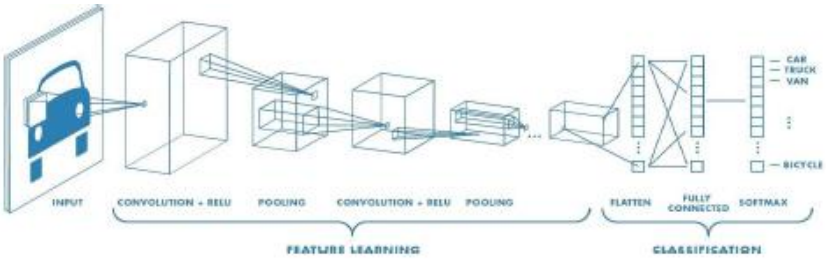
Gambar 2.1 Arsitektur *Multilayer* Perceptron

Menurut Haykin (2009) MLP (*Multilayer Perceptron*) dibangun dari 3 lapisan neuron yaitu, lapisan tersembunyi, lapisan masukan, dan lapisan keluaran. Masing-masing neuron pada lapisan keluaran dan lapisan tersembunyi terhubung dengan neuron pada lapisan sebelumnya melalui bobot (*weight*) yang dapat

disesuaikan sealam proses pembelajaran. MLP (*Multilayer Perceptron*) menggunakan algoritma *backpropagation* untuk mengatur bobot (*weight*) yang optimal sehingga dapat menghasilkan keluaran yang cocok dengan target yang diinginkan.

Jaringan saraf (neural networks) terdiri dari berbagai lapisan (layers), dan di setiap lapisan tersebut terdapat beberapa neuron yang saling terhubung dan memproses informasi secara hierarkis (Goodfellow, dkk, 2016). Multilayer Perceptron kurang handal dalam merumuskan persamaan bagi data yang lebih rumit. Dalam perkara jumlah lapisan tersembunyi dibawah tiga layer, telah ditemukan suatu pendekatan dalam menetapkan jumlah neuron pada setiap lapisan guna mencapai hasil yang sebaik mungkin. Seiring perkembangan deep learning, ditemukan kelemahan MLP dalam mengolah data yang rumit dapat ditanggulangi dengan suatu fungsi yang mengubah input menjadi bentuk yang lebih mudah dipahami. Atas dasar itu, berkembanglah jaringan saraf dengan tiga lapisan atau lebih.

Di tahun 1998, LeCun mengembangkan LeNet-5, sebuah jaringan konvolusional yang memiliki tujuh lapis (Haridas, dkk., 2019). Arsitektur LeNet-5 diterapkan guna memproses ekstraksi fitur antaramuka Brain Computer menggunakan Algoritma Common Spatial Patterns (CSPs) (Sakhavi, dkk., 2018). Namun, permasalahan utama yang dihadapi ialah akurasi rata-rata jaringan yang telah dimodifikasi justru lebih rendah dibandingkan akurasi model aslinya.

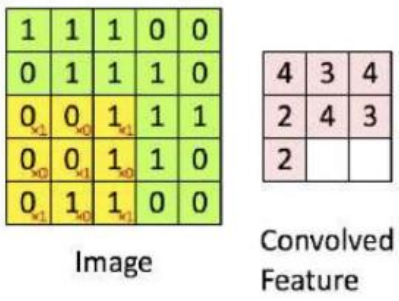


Gambar 2.2 Arsitektur *Convolutional Neural Network*

Berdasarkan Gambar 2.2, konsep fundamental pada *Convolutional Neural Network* (CNN) secara umum dikategorikan menjadi 4 komponen: *pooling layer, convolution layer, dropout*, dan *fully connected layer*. Lapisan-lapisan tersebut dimanfaatkan dalam menyusun arsitektur CNN yang sesuai untuk memproses masukan dalam bentuk citra atau gambar.

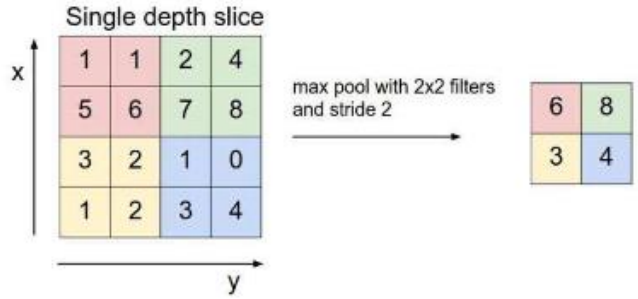
1) *Convolution Layer*

Convolution Layer merupakan lapisan inti dan paling esensial dalam metode CNN, yang bertanggung jawab dalam mengekstraksi fitur dari citra masukan dengan menghasilkan representasi bari dari citra tersebut. Lapisan ini terbentuk oleh rangkaian neuron yang menyusun filter dengan tinggi serta panjang. Gambar *Convolution Layer* bisa diamati pada berikut.



Gambar 2.3 Lapisan *Convolutional* CNN

Pooling layer ialah lapisan yang berfungsi memproses *feature maps* yang masuk dengan menggunakan operasi statistik terhadap nilai piksel yang letaknya berdekatan. Dalam struktur model CNN, *pooling layer* diatur berturutan untuk secara efisien mengurangi besar *output* pada *feature maps*. Hal ini mengurangi jumlah parameter serta perhitungan dalam jaringan, yang berfungsi menanggulangi *overfitting*. Sasaran utama dari *pooling* ialah mengecilkan dimensi *feature maps*, yang turut mempercepat proses penghitungan dengan mengurangi jumlah parameter yang harus diperbarui.



Gambar 2.4 *Pooling Layer*

2) *Fully Connected Layer*

Fully Connected Layer merupakan *layer* yang bertugas untuk menghubungkan aktivitas *layer* sebelumnya ke *layer* berikutnya, sebagaimana yang terjadi pada jaringan syaraf tiruan. Setiap aktivitas *neuron* harus ditransformasi ke dalam bentuk vektor satu dimensi untuk memungkinkan semua neuron pada *Fully Connected Layer* untuk saling terhubung.

3) *Dropout*

Dropout merupakan salah satu teknik regularisasi jaringan syaraf untuk mengecegah *overfitting*, dengan cara sejumlah neuron secara acak dinonaktifkan

selama pelatihan berlangsung. Oleh karena itu, kontribusi neuron-neuron yang nonaktifkan akan ditangguhkan sementara dan tidak ada bobot baru yang dimplementasikan pada *neuron* selama *backpropagation.*

c) Ragam Arsitektur CNN Modern

Selain komponen dasar, telah banyak dikembangkan arsitektur CNN spesifik yang dioptimalkan untuk berbagai kebutuhan, terutama untuk efisiensi pada perangkat seluler. Berikut adalah beberapa arsitektur modern yang sering digunakan dalam penelitian visi komputer

1) MobileNetV2

MobileNetV2 adalah arsitektur CNN yang dirancang khusus untuk aplikasi pada perangkat dengan sumber daya komputasi terbatas, seperti ponsel pintar. Arsitektur ini merupakan pengembangan dari versi sebelumnya dengan tujuan meningkatkan akurasi sekaligus menjaga efisiensi model (Sandler, dkk, 2018). Inovasi utama dalam MobileNetV2 adalah struktur blok *inverted residual with a linear bottleneck*. Struktur ini menggunakan lapisan konvolusi 1x1 untuk memperluas kanal, diikuti oleh konvolusi *depthwise*, dan diakhiri oleh lapisan konvolusi 1x1 (disebut *bottleneck*) yang dijaga tetap linier untuk mencegah hilangnya informasi. Hal ini menjadikan arsitektur lebih efisien dalam penggunaan memori dan komputasi (Sandler, dkk, 2018)

2) MobileNetV3Small

MobileNetV3 adalah generasi selanjutnya yang menawarkan efisiensi lebih tinggi. Arsitektur ini memanfaatkan pencarian arsitektur otomatis (*Neural Architecture Search*) untuk menemukan konfigurasi jaringan yang optimal dari segi

akurasi dan latensi (Howard, dkk, 2019). Varian MobileNetV3-Small dirancang secara spesifik untuk kasus penggunaan dengan sumber daya sangat rendah. Peningkatan utamanya meliputi integrasi modul *Squeeze-and-Excite* (SE) untuk mekanisme atensi fitur dan penggunaan fungsi aktivasi baru, *hard-swish* (h-swish), yang lebih efisien secara komputasi pada perangkat seluler (Howard, dkk, 2019)

3) EfficientNet-B0

EfficientNet memperkenalkan metode penskalaan model yang lebih sistematis dengan menyeimbangkan tiga dimensi jaringan (kedalaman, lebar, dan resolusi) secara bersamaan melalui metode *compound scaling* (Tan & Le, 2019). Model dasarnya, EfficientNet-B0, memiliki arsitektur yang ditemukan melalui *Neural Architecture Search* dan menggunakan blok *mobile inverted bottleneck* (MBConv) yang juga dipopulerkan oleh MobileNetV2. Berkat metode penskalaan yang seimbang, EfficientNet-B0 berhasil mencapai akurasi yang lebih tinggi dibandingkan model lain dengan ukuran serupa, namun dengan jumlah parameter dan kebutuhan komputasi yang jauh lebih sedikit (Tan & Le, 2019).

8. *Tensorflow*

Tensorflow lite merupakan pustaka *machine learning* yang dikembangkan oleh Google eksklusif bagi perangkat seluler. Dengan adanya hal ini, mesin dapat "belajar" pada perangkat dengan latensi rendah serta ukuran biner yang kecil.

Tensorflow sendiri adalah pustaka matematika simbolik yang dibuat didasarkan pada *dataflow* dan pemrograman diferensial. Pustaka ini digunakan oleh *Google* untuk keperluan penelitian maupun produksi.

Tensorflow Lite terdiri dari *runtime* yang berfungsi menjalankan model yang telah dikembangkan, serta serangkaian alat yang membantu mempersiapkan model untuk digunakan pada perangkat seluler dan perangkat yang terintegrasi.

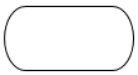
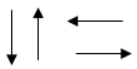
Sebelum mengembangkan model pelatihan. Model terlebih dahulu dilatiha pada mesin dengan daya komputasi tinggi, lalu mengkonversi model tersebut ke format .TFLITE, sebelum akhirknya dimuat ke dalam interpreter seluler untuk dieksekusi. (Widodo, B. 2016).

9. *Flowchart*

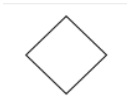
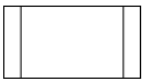
Menurut Arifianto (2014) *Flowchart* merupakan bagan yang memanfaatkan simbol khusus guna merepresentasikan rangkaian operasi serta keterkaitannya antara berbagai operasi dalam program.

Di bawah ini terdapat simbol-simbol yang dimanfaatkan dalam penyajian *flowchart*:

Tabel 2.1 Simbol *Flowchart*

Simbol	Nama dan Fungsinya
	***Terminal Point Symbol* / Simbol Titik Terminal** Menandakan titik awalan (start) atau titik akhiran (stop) suatu operasi.
	***Flow Direction Symbol* / Simbol Arus** Dipergunakan demi mengaitkan sebuah simbol dengan simbol lainnya (connecting line) serta menandakan arah garis alir proses.
	***Processing Symbol* / Simbol Proses** Simbol ini mengindikasikan aktivitas pemrosesan yang dilakukan oleh komputer. *Processing simbol* melambangkan aktivitas pemeriksaan atau yang sering disebut sebagai simbol pemeriksaan dalam bidang industri (proses produksi barang).

Lanjutan Tabel 2.1

Simbol	Nama dan Fungsinya
	***Decision Symbol* / Simbol Keputusan** Dipergunakan guna menentukan operasi atau decision berlandaskan keadaan.
	**I*nput-Output* / Simbol Keluar-Masuk** Memberitahukan operasi masukan serta keluaran (*input dan output*) yang berlangsung secara independen terhadap perangkat yang digunakan.
	***Predefined Process* / Simbol Proses Terdefinisi** Mengindikasikan penerapan atau implementasi suatu prosedur (sub-process) yang telah ditetapkan sebelumnya.
	***Connector (On-page)*** Dipergunakan untuk merapikan relasi antara simbol-simbol yang terpisah jauh dalam satu halaman yang sama, sehingga mengurangi kompleksitas alur proses yang digambarkan.
	***Connector (Off-page)*** Serupa dengan *on-page connector*, tetapi simbol ini dipergunakan guna mengaitkan simbol proses yang berada pada halaman berbeda. Angka atau huruf adalah label yang dapat digunakan untuk simbol ini.
	***Preparation Symbol* / Simbol Persiapan** Mengindikasikan proses menyiapkan sistem media penyimpanan data atau *storage*.
	***Manual Input Symbol*** Menampilkan proses masukan data ala manual, misalkan melalui papan ketik.
	***Manual Operation Symbol* / Simbol Kegiatan** Berfungsi untuk menampilkan kegiatan/proses yang dilakukan secara manual tanpa keterlibatan komputer.
	***Document Symbol*** Berfungsi untuk merepresentasikan dokumen sebagai sumber data masukan atau keluaran yang harus dicetak dalam bentuk kertas.
	***Multiple Documents*** Serupa dengan *document symbol*, akan tetapi jumlah dokumen yang digunakan lebih dari satu.

Lanjutan Tabel 2.1

Simbol	Nama dan Fungsinya
	***Display Symbol*** Mendeklarasikan penggunaan peranti keluaran semacam monitor printer, monitor, serta alat serupa lainnya.
	***Delay Symbol*** Digunakan untuk menampilkan operasi delay (penundaan) dalam sistem, semacam proses menunggu surat untuk diarsipkan atau antrian eksekusi suatu tugas

10. UML *(Unified Modelling Language)*

Unified Modelling Language (UML) merupakan "bahasa" yang telah menjadi pedoman industri guna mendesain, menvisualisasikan, serta pengarsipan sistem peranti lunak. *Unified Modelling Language* (UML) menyediakan kerangka umum guna mendesain model sistem (Sora, 2015).

Unified Modelling Language (UML) sebagai perkakas guna merancang model sistem atau piranti lunak berbasis objek, seraya menunjang keberlanjutan dalam pengembangannya. Ketidakadaan dokumentasi yang mumpuni akan menghalangi proses pengembangan karena developer wajib memeriksa, menelusuri, serta mempelajari kode program. Berkat UML, pertukaran pengetahuan terkait aplikasi ataupun sistem yang tengah dibangun dapat dilaksanakan dari satu pengembang ke pengembang lain. Bahkan, bukan cuma kalangan pengembang, tetapi juga memudahkan orang bisnis dan siapa saja dalam memahami suatu sistem.

Pada Januari 1997, Objek Management Group merancang UML serta merilis versi 1.0. Dalam pendekatan berbasis objek, adad sejumlah prinsip yang

wajib diperhatikan yaitu: *Class, Encapsulation, Object, Inheritance, Abstraction* serta *Polymorphism*. UML memiliki berbagai jenis diagram, di antaranya:

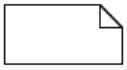
a) *Use Case Diagram*

Fokus utama dari *Use Case Diagram* ialah menunjukkan fungsionalitas sistem yang diharapkan, dengan menyoroti "apa" yang dilakukan sistem seraya mengesampingkan "bagaimana" cara kerjanya. Pada diagram, *Use Case* menerangkan relasi aktor dengan sistem. *Use Case* tersebut melambangkan tugas spesifik, misalnya membuat sebuah daftar belanja, login ke sistem, dan lain sebagainya. Entitas manusia maupun mesin yang berinteraksi dengan sistem guna melaksanakan tugas tertentu dikenal sebagai aktor dalam *Use Case Diagram*. Berikut simbol *Use Case Diagram*:

Tabel 2.2 Simbol Use Case Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		*Actor*	Menunjukkan bagaimana pengguna berperan dalam melakukan interaksi dengan *Use Case*.
2		*Dependency*	Menerangkan keterkaitan di mana perubahan elemen independen dapat berdampak pada elemen non-independen.
3		*Generalization*	Merepresentasikan keterkaitan hirarki, di mana objek anak mewarisi struktur data serta perilaku objek induknya.
4		*Include*	Mengisyarakatkan jika suatu *Use Case* mencakup perilaku dari *Use Case* lainnya secara eksplisit bagian dari prosesnya.

Lanjutan Tabel 2.2

NO	GAMBAR	NAMA	KETERANGAN
5		*Extend*	Menggambarkan jika suatu *Use Case* menambahkan *behavior Use Case* lain pada bagian tertentu.
6		*Association*	Mendefinisikan keterhubungan dua objek atau lebih dalam suatu sistem.
7		*System*	Menentukan komponen atau paket yang hendak ditampakkan sistem dengan batasan tertentu.
8		*Use Case*	Mendeskripsikan serangkaian proses-proses yang dikerjakan sistem guna menghasilkan output yang dapat diamati oleh aktor.
9		*Collaboration*	Mewakili interaksi antara berbagai elemen dalam sistem yang menghasilkan perilaku kolektif yang lebih besar daripada jumlah bagian individunya (sinergi).
10		*Note*	Elemen fisik yang hadir dalam kala aplikasi beroperasi serta melambangkan sumber data komputasi.

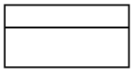
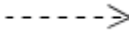
b) *Class Diagram*

Adapun simbol-simbol *Class Diagram* antara lain :

Tabel 2.3 Simbol *Class* Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		*Generalization*	Merepresentasikan keterkaitan hirarki, di mana objek anak mewarisi struktur data serta perilaku objek induknya.
2		*Nary Association*	Hubungan yang mengaitkan 3 atau lebih objek, yang sepatutnya dijauhi.

Lanjutan Tabel 2.3

NO	GAMBAR	NAMA	KETERANGAN
3		*Class*	Merepresentasikan kumpulan objek degnan sifat dan cara kerja serupa.
4		*Collaboration*	Mewakili interaksi antara berbagai elemen dalam sistem yang menghasilkan perilaku kolektif yang lebih besar daripada jumlah bagian individunya (sinergi).
5		*Realization*	Proses implementasi nyata dari suatu operasi dalam sebuah objek.
6		*Dependency*	Menerangkan keterkaitan di mana perubahan elemen independen dapat berdampak pada elemen non-independen.
7		*Association*	Mendefinisikan keterhubungan dua objek atau lebih dalam suatu sistem.

c) *Sequence Diagram*

Adapun simbol-simbol *Sequence* Diagram antara lain :

Tabel 2.4 Simbol *Sequence* Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		*LifeLine*	Representasi dari Objek *entity*, atau antaramuka yang saling berhubungan dalam sistem.
2		*Message*	Menunjukkan perincian komunikasi antara objek yang membawa informasi terkait proses yang berjalan.
3	S	*Message*	Perincian komunikasi antara objek yang membawa informasi terkait proses yang berjalan.

d) *StateChart Diagram*

Adapun simbol-simbol *StateChart* Diagram antara lain :

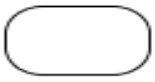
Tabel 2.5 Simbol *StateChart* Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		*State*	Menunujukkan status suatu objek pada titik waktu tertentu. Ini termasuk nilai atribut dan nilai hubungan (tautan).
2		*Initial Pseudo State*	Menentukan titik awalan atau status awal saat objek pertama kali dibuat.
3		*Final State*	Menunjukkan keadan akhir suatu objek, yaitu keadaan saat objek tersebut dihancurkan atau tidak lagi digunakan.
4		*Transition*	Suatu peristiwa yang merubah satu atau lebih nilai atribut dari suatu objek.
5		*Association*	Mendefinisikan keterhubungan dua objek atau lebih dalam suatu sistem.
6		*Node*	Elemen fisik yang hadir dalam kala aplikasi beroperasi serta melambangkan sumber data komputasi.

e) *Activity Diagram*

Adapun simbol-simbol *Activity Diagram* antara lain :

Tabel 2.6 Simbol *Activity* Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		*Activity*	Mengilustrasikan relasi antar kelas antaramuka dalam sistem.
2		*Action*	Mewakili status sistem yang menjelaskan implementasi proses tertentu.

Lanjutan Tabel 2.6

NO	GAMBAR	NAMA	KETERANGAN
3		*Initial Node*	Menunjukkan titik awal ataupun proses inisialisasi suatu objek dalam diagram.
4		*Activity Final Node*	Menunjukkan titik awal atau proses pembuatan atau penghancurkan objek.
5		*Fork Node*	Menunjukkan titik percabangan tempat aliran proses tunggal terbagi menjadi beberapa aliran.

11. Pengujian *Whitebox*

Menurut Harya, dkk. (2022) *White box testing* atau pengujian kotak putih adalah pengujian yang menganalisis kode program untuk mendeteksi adanya kesalahan atau tidak. Sedangkan, Menurut Rex Black (2009:128) *white box testing* ialah metode pengetesan piranti lunak (*software*) dengan memanfaatkan struktur internal program dalam pembuatan kasus uji. Pendekatan ini dapat diterapkan untuk mengidentifikasi kesalahan dalam logika program, seperti keputusan yang salah, kondisi yang hilang, serta batas kasus.

Berlandaskan interpretasi-interpretasi sebelumnya, peneliti menyimpulkan bahwa *white box testing* ialah metodologi pengujian piranti lunak (*software*) yang menitikberatkan di kode program. Untuk memverfikasi program berfungsi dengan benar, pengujian ini esensial dilaksanakan.

Rex Black (2009:129) mengemukakan bahwa ada beberapa kelebihan dan kekurangan dari *white box testing*, sebagaimana tertera berikut keunggulan serta kelemahan dari white box testing:

a) Kelebihan *white box testing*

1) Mampu mengidentifikasi kesalahan logika yang tidak terdeteksi oleh pengujian *black box*. Metodologi pengetesan *black box* hanya menitikberatkan di fungsionalitas keluaran program, sedangkan pengujian *white box* lebih menitikberatkan pada struktur internal kode program.
2) Membantu meningkatkan kinerja dan efisiensi program. Dengan memahami struktur internal kode program, tester dapat mengindentifikasi area yang dapat ditingkatkan kinerja dan efisiensinya.
3) Membantu mengurangi biaya perawatan program.

b) Kekurangan *white box testing*

1) Sangat kompleks dan memakan waktu, terutama untuk program yang besar dan kompleks. Pengujian white box memerlukan pemahaman yang mendalam tentang struktur internal kode program.
2) Membutuhkan pengetahuan mendalam tentang struktur internal kode program.

Ada beberapa teknik white box testing yang umum digunakan menurut Saefudin (2022:110), antara lain yaitu:

a) Metode Basis Path

Metode basis path adalah metode yang bertujuan untuk menguji semua halur eksekusi yang mungkin terjadi dalam program. Teknik ini menggunakan algoritma untuk menghitung jumlah jalur eksekusi yang mungkin terjadi, dan kemudian

menghasilkan Kumpulan kasus uji yang dapat menguji semua jalur eksekusi tersebut.

b) Metode Kontrol Aliran

Metode ini bertujuan untuk menguji semua kondisi yang mungkin terjadi dalam program. Teknik ini menggunakan diagram control aliran untuk mengidentifikasi semua kondisi yang mungkin terjadi, dan kemudian menghasilkan Kumpulan kasus uji yang dapat menguji semua kondisi tersebut.

c) Metode Data

Metode ini berfokus pada data inpunt dan ouput program. Teknik ini bertujuan untuk memastikan bahwa program dapat menangani berbagai jenis data input dengan benar.

12. Pengujian *Blackbox*

Jafar, dkk. (2021) berpendapat bahwa pengujian *black box* ialah metodologi uji yang bertitikberat di perincian fungsional peranti lunak. Sedangkan, menurut Rizky, dkk. (2020), pengujian *black box* berperan sebagai pengujian pelengkap terhadap pengujian sebelumnya.

Berdasarkan berbagai definisi di atas, peneliti mengambil kesimpulan yakni pengujian *black box* ialah metodologi pengujian yang efektif dalam mendeteksi cacat perangkat lunak, khususnya yang berkaitan dengan antarmuka.

Menurut Saefudin (2022:110), pengujian black box mempunyai sejumlah keunggulan serta kelemahan yaitu:

a) Kelebihan Pengujian *Black Box*

1) Mudah dipahami dan diterapkan

2) Dapat digunakan untuk menguji semua aspek perangkat lunak

3) Efektif untuk menemukan kesalahan yang berhubungan dengan antarmuka

b) Kekurangan Pengujian *Black Box*

1) Tidak dapat menemukan kesalahan yang berhubungan dengan struktur internal perangkat lunak

2) Dapat menghasilkan sejumlah besar *test cas*

B. Tinjauan Hasil Penelitian Terdahulu

Dengan menganalisis berbagai judul dan topik penelitian yang sedang berlangsung, dapat diidentifikasi kesamaan serta perbedaan dari penelitian yang akan dilakukan. Analisis ini bertujuan untuk membuktikan bahwa keaslian penelitian serta menghindari duplikasi terhadap tugas akhir dari penelitian lain, seperti berikut ini:

1. Febriani. F. D. (2020). "Klasifikasi Citra Kue Tradisional Indonesia Berdasarkan Ekstraksi Fitur Warna RGB Color Moment dan K-Nearest Neighbor" Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, 3(10), 10199-10206. Dalam penelitian ini, sebanyak 29 kue diklasifikasikan ke dalam 29 kategori yang terdiri dari delapan, lima, dan tiga kategori. Dengan mengkombinasikan fitur *Color Moment* dengan metode KNN telah terbukti menunjukkan nilai evaluasi terbaik yakni 60% di ketiga kategori dalam skenario pengujian.

2. Ilahiyah, S., & Nilogiri, A. (2018). "Implementasi Deep Learning Pada Identifikasi Jenis Tumbuhan Berdasarkan Citra Daun Menggunakan Convolutional Neural Network" Jurnal Teknologi Informasi, 3(02), 49-56. Berdasarkan hasil

pengamatan peneliti diperoleh rata-rata akurasi berkisar 85,21% sampai dengan 90,8% memanfaatkan *K-Fold Cross Validation* dengan parameter k=10.

3. Shafira, T. (2018). "Implementasi Convolutional Neural Networks Untuk Klasifikasi Citra Tomat Menggunakan Keras" (Tugas Akhir, Program Studi Statistika, Universitas Islam Indonesia Yogyakarta). Hasil yang diperoleh menunjukkan tingkat akurasi model yang dikembangkan pada data uji sebesar 90% dalam mengklasifikasikan gambar tomat, menunjukkan kemampuannya dalam mengindentifikasi buat tomat dengan baik.

4. Yudianto, M. R. A., Kusrini, & Al Fatta, H. (2020). "Analisis Pengaruh Tingkat Akurasi Klasifikasi Citra Wayang Dengan Algoritma Convolutional Neural Network" Jurnal Teknologi Informasi, 4(02), 182-190. Dataset yang terdiri dari 400 gambar wayang punakwan digunakan dalam penelitian ini. Pada percobaan yang dilakukan, presisinya sebesar 93% dan *recall*-nya sebesar 87%, dengan hasil akurasi terbaik dicapai sebesar 97%.

5. Ridwan, N., Andryana, S., & Winarsih. (2018). “Pengenalan aksara Lontara tulis tangan menggunakan metode convolutional neural networks berbasis android”. Jurnal Informatika, 18(1), 93-1011. Penelitian ini mengembangkan aplikasi Android untuk mengenali aksara Lontara tulis tangan menggunakan metode CNN. Dataset dimanfaatkan dan berisi 30 karakter Lontara yang ditulis dalam delapan font berbeda. Dilatih menggunakan pustaka TensorFlow, aplikasi ini mencapai akurasi 90% dalam mengenali bentuk karakter Lontara individu.

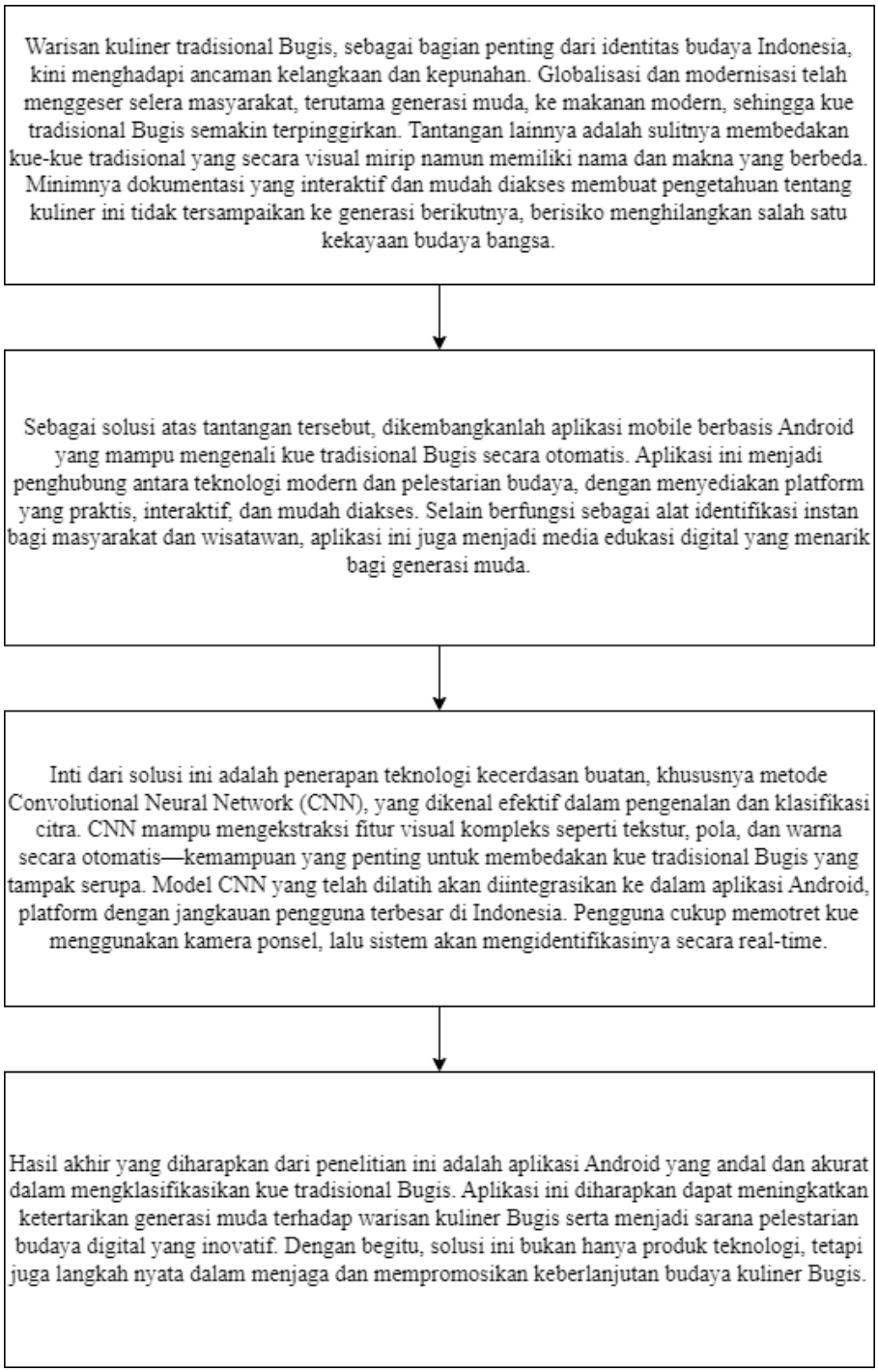
Dari 5 penelitian yang telah dilaksanakan, terlihat perbedaan-perbedaan signifikan dengan kajian-kajian terdahulu. Penelitian terdahulu, seperti yang

dilakukan oleh Febriani, Ilahiyah, dan Nilogiri, Shafira, serta Yudianto, Kusrini, dan Al Fatta, membahas berbagai konteks seperti klasifikasi citra kue tradisional, identifikasi jenis tumbuhan berdasarkan citra daun, klasifikasi citra tomat, dan klasifikasi citra wayang.

Sebagai kontrast, penelitian ini fokus pada pengembangan aplikasi Android dengan fitur deteksi kue tradisional Bugis menggunakan metode *Convolutional Neural Network* (CNN). Perbedaan utama terletak di objek penelitian yang dipilih, pendekatan metode deteksi, dan tujuan aplikatif penggunaan teknologi deteksi visual. Sehingga, penelitian ini memberikan kontribusi baru pada eksplorasi aplikatif teknologi deteksi visual untuk mengenali keberagaman kue tradisional Bugis.

C. Kerangka Pikir

Untuk lebih memperjelas kerangka pemikiran, diagram berikut dapat digunakan sebagai ilustrasi:



Gambar 2.5 Kerangka Pikir

BAB III

METODE PENELITIAN

A. Lokasi dan Waktu Penelitian

Penelitian dilaksanakan di Universitas Muhammadiyah Parepare, Jl. Jend. Ahmad Yani Km. 6, Bukit Harapan, Kec. Soreang, Kota Parepare, Provinsi Sulawesi Selatan 91112 sepanjang tiga 3 bulan, bermula dari bulan ketiga hingga bulan kelima tahun 2024.

Tabel 3.1 Jadwal Pelaksanaan Penelitian

No.	Uraian Kegiatan	Bulan 2024		
		Maret	April	Mei
1	Studi Literatur			
2	Pembuatan Aplikasi			
3	Pengujian Aplikasi			
4	Hasil Pengujian			

B. Jenis Penelitian

Penelitian ialah ikhtiar yang dijalankan dengan kesengajaan demi mengenali atau mengkaji fakta-fakta baru, serta untuk mencari bukti-bukti relevan dengan suatu masalah. Menurut Dini & Tuti (2020), penelitian juga dapat dipahami sebagai penerapan pendekatan ilmiah pada dalam menganalisis masalah dengan tujuan memperoleh informasi yang bermanfaat dan dapat dipertanggungjawabakan.

Setelah mendapatkan data secara studi pustaka, proses penelitian akan dilakukan eksperimen atau percobaan. Dalam eksperimen ini pengumpulan data

dapat diambil secara langsung, sehingga akan lebih mendalami dalam melakukan penelitian. Penelitian juga meminimalkan kesalahan dalam penelitian ini.

C. Metode Pengumpulan Data

Pengumpulan data diadakan guna menghimpun semua informasi relevan serta mendorong pelaksanaan penelitian ini.

1. Pengamatan Langsung (Observasi), Teknik ini dilakukan melalui pengamatan langsung terhadap proses pembuatan kue tradisional Bugis, dengan tujuan yaitu mendapatkan gambaran tentang bahan, alat, cara dan hasil pembuatan kue tradisional Bugis.
2. Kajian Kepustakaan, metode pengumpulan data yaitu melalui upaya menelusuri serta mempelajari sumber-sumber literatur berkaitan tentang kue tradisional Bugis seperti buku resep, artikel, jurnal dengan tujuan mendapatkan informasi tentang jenis, resep, dan nilai kue tradisional bugis.
3. Dokumentasi, metode ini dilakukan dengan mengumpulkan dan mendokumentasikan gambar-gambar kue tradisional Bugis yang digunakan menjadi data masukan untuk model *Convolutional Neural Network* (CNN). Gambar-gambar tersebut diperoleh dari beragam sumber seperti internet, buku, atau dokumentasi pribadi.

D. Alat dan Bahan Penelitian

Penelitian ini memanfaatkan sekalian perkakas serta bahan, yang mana meliputi barang keras (Hardware) pun piranti lunak (Software). Adapun barang keras yang difungsikan dalam kajian ini, antara lain Personal Computer (PC) atau

laptop merk ASUS, sementara perangkat lunak nan digunakan ialah Windows 10 x64 serta Android Studio. Sedangkan dataset penelitian diperoleh dari melalui pengambilan gambar kue tradisional Bugis secara langsung dengan menggunakan kamera ponsel pintar(Smartphone), dan beberapa gambar tambahan diperoleh dari Google Images.

E. Tahap Penelitian

Tahap penelitian dilakukan dengan cara :

1. Persiapan Penelitian

Peneliti mempersiapkan segala hal yang diperlukan untuk menunjang penelitian, termasuk menyiapkan referensi seperti buku-buku, artikel yang berasosiasi dengan topik penelitian, pun demikian perangkat lunak yang dimanfaatkan sepanjang masa penelitian.

2. Pengumpulan Data

Pada tahap ini, peneliti mengumpulan informasi terkait kue tradisional bugis, yang mencakup aspek-aspek seperti nama kue, deskripsi singkat tentang kue, komposisi bahan, prosedur pembuatan, dan gambar-gambar kue tradisional bugis. Selain itu, peneliti juga menyusun dataset untuk model *Convolutional Neural Network* (CNN) dengan mengumpulkan beragam gambar kue tradisional bugis dari berbagai sumber, termasuk internet, buku, dan dokumentasi pribadi.

3. Analisis

Peneliti menganalisis data yang tersedia serta sistem, lalu merumuskan permasalahan utama yang menjadi titik berat penelitian, hingga ditemukan kaidah penyelesaikan yang patut.

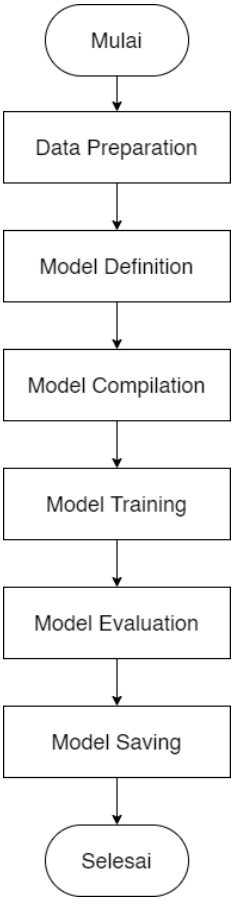
4. Perancangan

Peneliti merancang atau mendesain sistem pengenalan kue tradisional Bugis berbasis Android. Aspek-aspek perancangan melibatkan tata letak *user interface* (UI), fungsionalitas aplikasi, integrasi teknologi pengenalan gambar untuk mencapai tujuan utama pengembangan aplikasi ini. Salah satu fitur aplikasi ini adalah deteksi kue tradisional Bugis, yang mengimplementasikan metode *Convolutional Neural Network* (CNN). Pengembangan sistem fitur tersebut dilakukan dengan menggunakan Bahasa Pemrograman *python*, dengan dukungan dari infrastruktur Google Colab. Model hasil pelatihan di Google Colab kemudian diintegrasikan ke dalam Android Studio menggunakan TensorFlow Lite. Kemudian, aplikasi dikembangkan menggunakan bahasa pemrograman Java dan Kotlin, dan dikemas menjadi sebuah file aplikasi dengan ekstensi .apk. Proses pembuatan aplikasi ini didasarkan pada alternatif pemecahan masalah.

5. Tahapan Pembuatan Model

Pada tahap ini, dapat digambarkan dalam bentuk *flowchart*. *Flowchart* adalah representasi grafis yang menunjukkan urutan serta keterkaitan antar proses dan instruksi. Setiap proses direpresentasikan melalui simbol tertentu, sementara hubungan antarproses digambarkan menggunakan garis penghubung. Berikut *flowchart* pembuatan model CNN:

a. Flowchart pembuatan model CNN



Gambar 3.1 *Flowchart* Pembuatan Model

Proses dimulai dengan menentukan tujuan klasifikasi dan mengumpulkan dataset gambar kue tradisional Bugis, yang kemudian melalui tahap Data Preparation berupa preprocessing, normalisasi, serta pemisahan dataset menjadi data training, validation, dan testing. Selanjutnya, dilakukan tahap Model Definition dengan memanfaatkan tiga arsitektur model Convolutional Neural Network (CNN) yaitu MobileNetV2, MobileNetV3Small, dan EfficientNetB0 yang telah dilatih sebelumnya (pre-trained) pada dataset ImageNet. Model-model ini digunakan melalui pendekatan transfer learning, di mana seluruh lapisan dari model dasar (base model) dibekukan (tidak dilatih ulang) agar proses pelatihan menjadi

lebih cepat dan stabil, serta tetap dapat memanfaatkan fitur-fitur visual umum yang telah dipelajari oleh model dari dataset besar. Penyesuaian dilakukan dengan menambahkan fully connected layer pada bagian akhir arsitektur, yaitu dengan menambahkan lapisan Global Average Pooling, Dense, Dropout, dan output layer dengan fungsi aktivasi softmax sesuai jumlah kelas kue tradisional.

Model kemudian dikompilasi (Model Compilation) menggunakan optimizer Adam dengan learning rate sebesar 0.0001, fungsi loss categorical crossentropy, serta metrik akurasi. Pada tahap Model Training, model dilatih menggunakan data training dan dievaluasi melalui data validation, sedangkan Model Evaluation dilakukan menggunakan data testing untuk mengukur akurasi serta metrik performa lainnya. Setelah diperoleh model terbaik, Model Saving dilakukan dengan menyimpan model tersebut dan mengonversinya ke format TFLite agar siap digunakan dalam aplikasi deteksi kue tradisional Bugis. Proses ini diakhiri dengan tahap implementasi model ke dalam sistem aplikasi Android.

6. Pengujian

Melakukan eksperimen menggunakan metode yang digunakan untuk mengevaluasi output yang diperoleh dari data penelitian. Jika desain yang dihasilkan mengungkapkan cacat atau kelemahan, prosedur analisis diulang. Kami kemudian mengintegrasikan sistem yang dibuat ke dalam aplikasi Android menggunakan metode *Convolutional Neural Network* (CNN).

7. Implementasi

Aplikasi siap untuk digunakan, setelah tidak ditemukan kekurangan pada tahap ini.

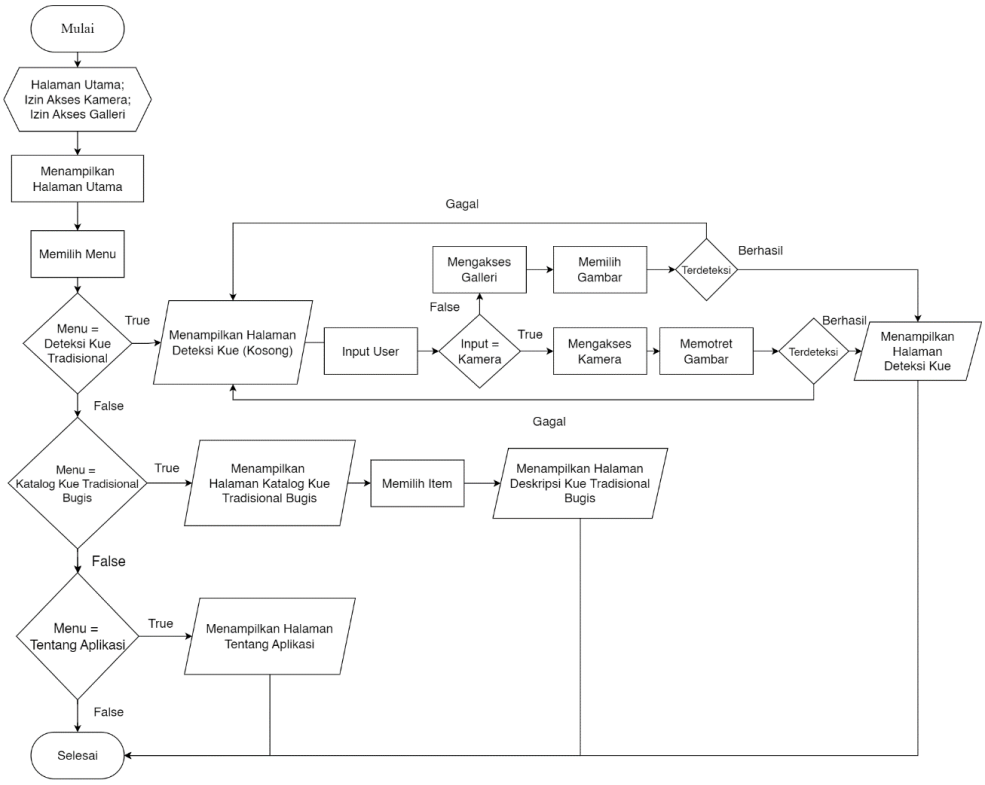
F. Metode Pengujian

Pengujian sistem yang dikembangkan diimplementasikan dalam bahas pemrograman Java dan Python dilakukan berdasarkan mekanisme yang dirancang dalam rencana pengujian dan diklasifikasikan berdasarkan jenis pengguna. Proses pengujian menggunakan dua metode utama untuk mengevaluasi setiap fungsi dalam sistem.

White Box ialah pengujian pertama, berpusat di analisis struktur internal *software*, seperti proses pengulangan (*looping*) dan prosedur program (*basis path*). Berikutnya adalah pengujian *Black Box*, yang mengevaluasi fungsionalitas perangkat lunak dengan mengamati hasil eksekusi berdasarkan data uji tanpa mempertimbangkan struktur internalnya.

G. Analisa Sistem

1. Flowchart Sistem

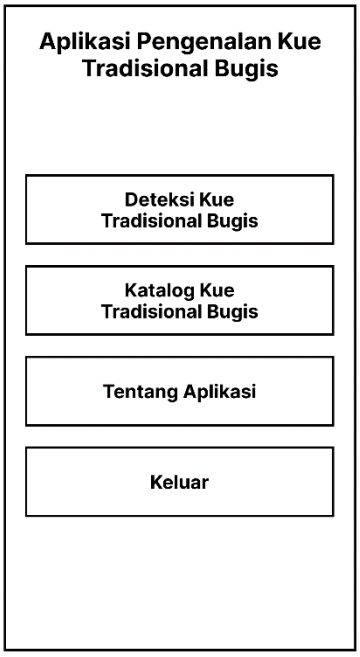


Gambar 3.2 *Flowchart* Aplikasi Android

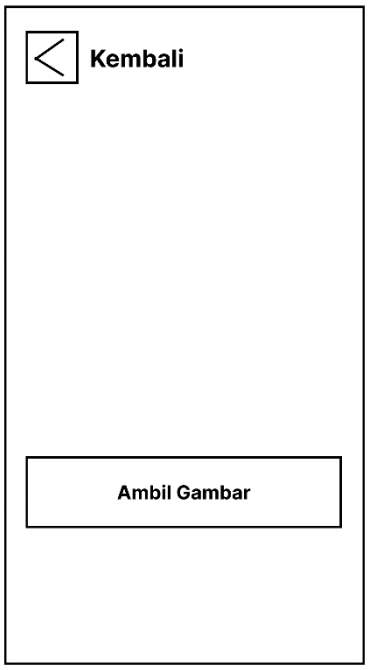
Flowchart tersebut mengilustrasikan langkah-langkah dalam proses pengambilan gambar dan menampilkan katalog kue yang sesuai dengan gambar yang terdeteksi. Proses dimulai dengan mengambil gambar melalui kamera atau galeri, yang kemudian diperiksa oleh model CNN yang telah dibuat sebelumnya. Apabila model CNN gagal dalam pengecekan gambar, proses kembali ke tahap pengambilan gambar. Namun, jika model CNN berhasil mengklasifikasikan gambar, teks berupa hasil klasifikasi gambar kue tradisional Bugis bersama dengan gambar kue tradisional Bugis akan ditampilkan di layar smartphone. Selanjutnya, hasil klasifikasi tersebut akan diperiksa untuk memastikan apakah kue tersebut ada

dalam katalog. Jika ada, pengguna dapat beralih ke katalog kue tradisional Bugis. Namun, jika gambar tidak sesuai dengan katalog, proses berakhir dan keluar. Dengan kata lain, flowchart ini merinci proses pengambilan gambar, menampilkan hasil klasifikasi, dan memverifikasi keberadaan hasil klasifikasi dalam katalog Kue Tradisional Bugis. Jika sesuai, katalog akan ditampilkan; sebaliknya, proses akan berakhir. Langkah-langkah ini dapat bervariasi tergantung pada kebutuhan aplikasi dan implementasi model yang digunakan.

2. UI (User Interface)



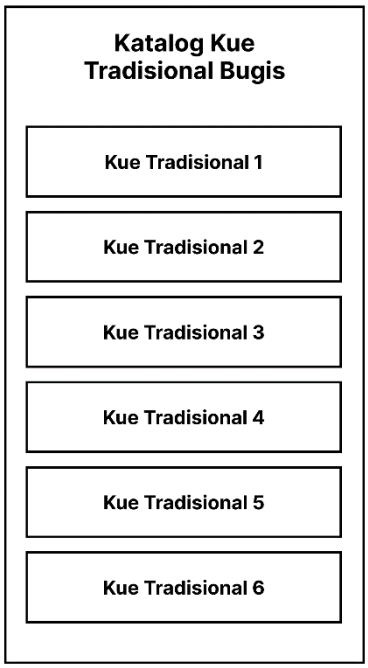
Gambar 3.3 Antarmuka *Home Page*



Gambar 3.4 Antarmuka Halaman Deteksi Kue Tradisional Bugis



Gambar 3.5 Antarmuka Halaman Hasil Deteksi Kue Tradisional Bugis



Gambar 3.6 Antarmuka Halaman Katalog Kue Tradisional Bugis



Gambar 3.7 Antarmuka Halaman Deskripsi Kue Tradisional Bugis



Gambar 3.8 Antarmuka Halaman Tentang Aplikasi

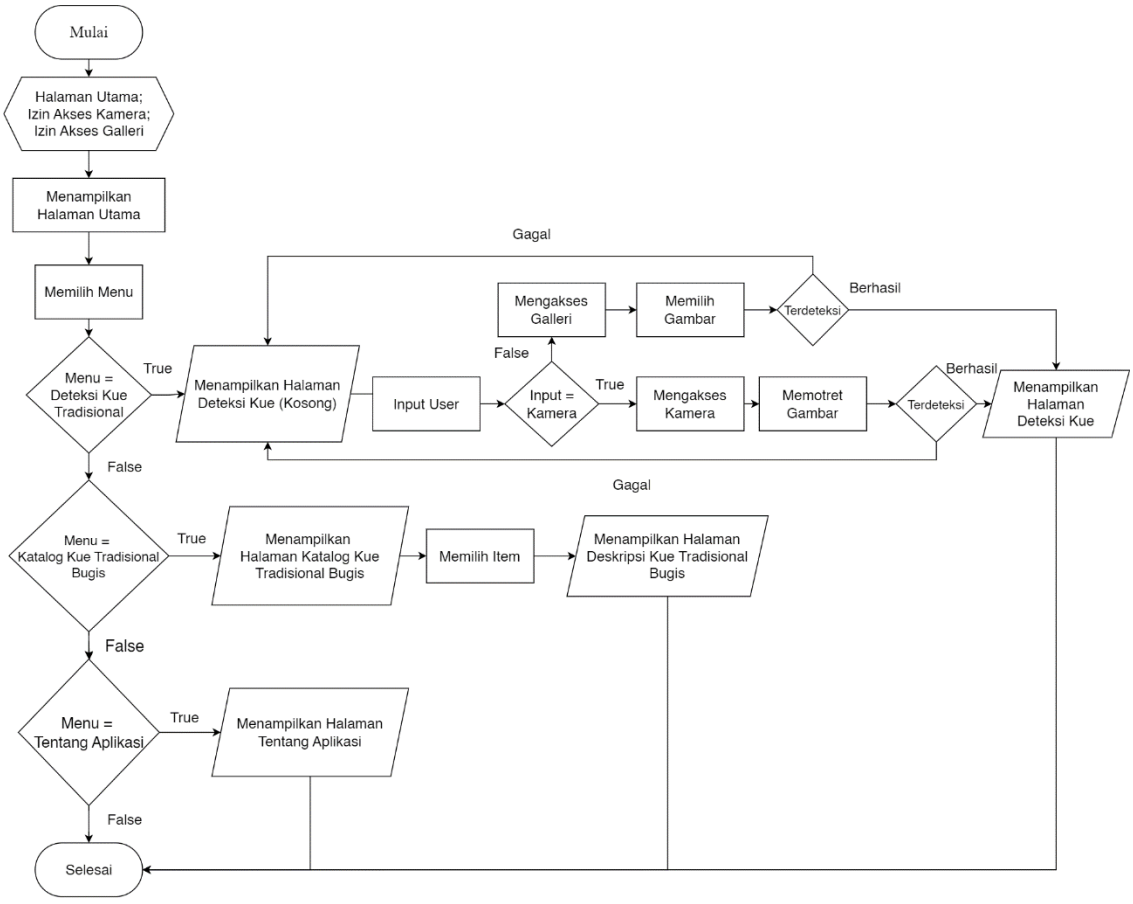
BAB IV

HASIL DAN PEMBAHASAN

A. Analisis Sistem Yang Diusulkan

Diajukan sebuah sistem dalam bentuk aplikasi Android yang berbasis *Convolutional Neural Network* (CNN) untuk mengenali jenis kue tradisional Bugis. Aplikasi ini menganalisis gambar kue dan menampilkan hasil deteksi. Aplikasi ini juga dilengkapi dengan fitur katalog yang menampilkan berbagai jenis kue tradisional Bugis, memberikan informasi tentang masing-masing jenis kue.

1. *Flowchart Diagram* Aplikasi

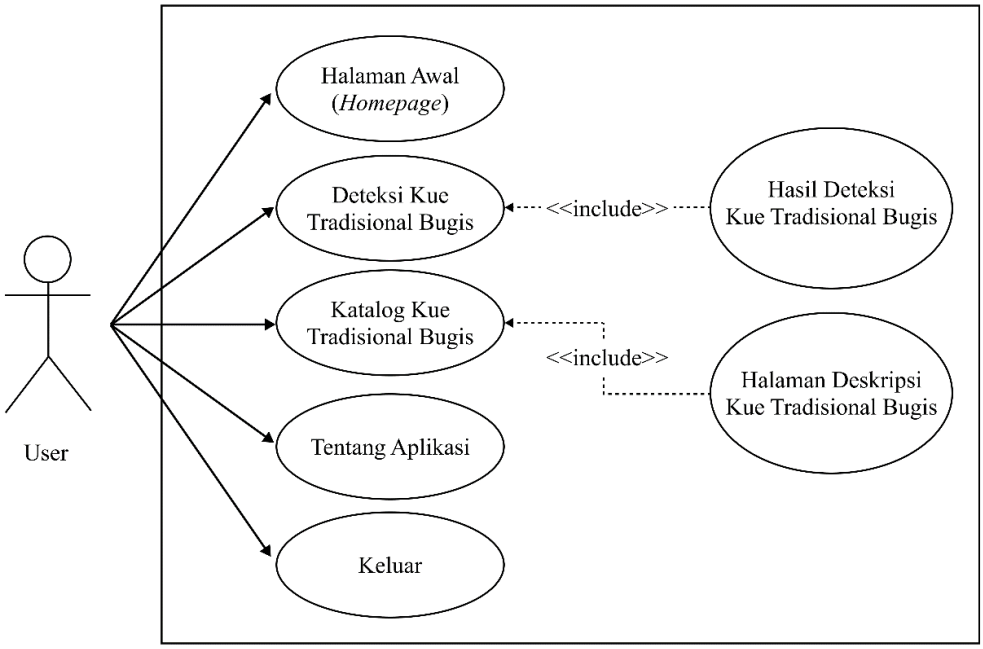


Gambar 4.1 *Flowchart Diagram*

Flowchart ini menggambarkan alur kerja sebuah aplikasi yang memiliki tiga fitur utama: deteksi kue tradisional, katalog kue tradisional Bugis, dan informasi tentang aplikasi. Proses dimulai dengan membuka aplikasi, di mana pengguna diminta memberikan izin akses ke kamera dan galeri. Setelah izin diberikan, aplikasi menampilkan halaman utama. Pengguna kemudian memilih salah satu dari tiga menu yang tersedia. Jika memilih menu Deteksi Kue Tradisional, aplikasi akan menampilkan halaman deteksi dalam keadaan kosong. Pengguna dapat mengunggah gambar melalui galeri atau mengambil foto menggunakan kamera. Setelah gambar tersedia, sistem akan mencoba mendeteksi kue dalam gambar tersebut. Jika deteksi berhasil, aplikasi akan menampilkan halaman hasil deteksi, tetapi jika gagal, pengguna dapat mencoba lagi.

Jika pengguna memilih menu Katalog Kue Tradisional Bugis, aplikasi akan menampilkan daftar katalog kue yang tersedia. Pengguna kemudian dapat memilih salah satu item untuk melihat deskripsi lengkapnya. Jika terjadi kegagalan dalam menampilkan deskripsi, sistem akan kembali ke katalog. Sementara itu, jika pengguna memilih menu Tentang Aplikasi, aplikasi akan menampilkan informasi mengenai aplikasi tersebut. Jika pengguna tidak memilih menu apa pun, proses akan selesai.

2. *Use Case Diagram*



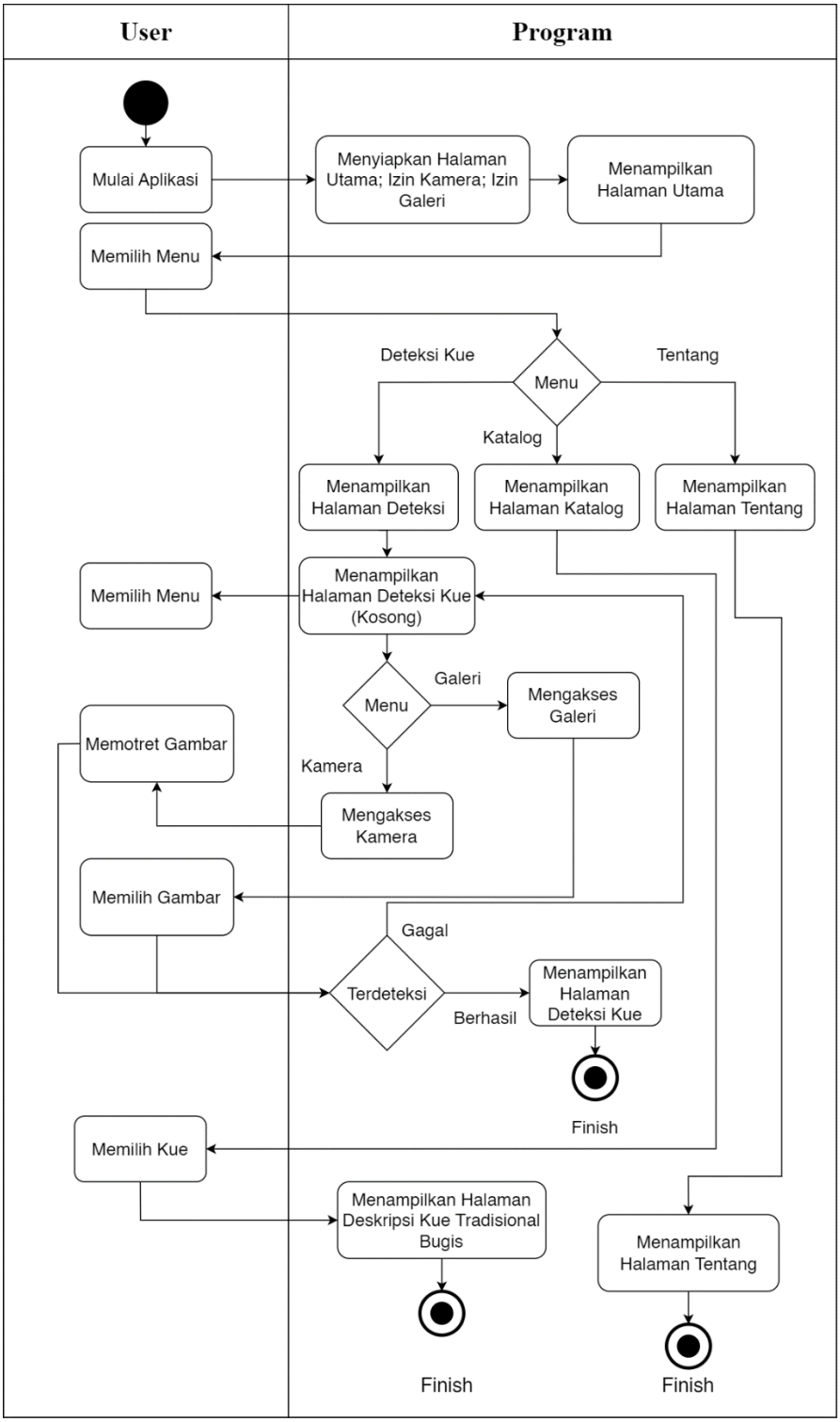
Gambar 4.2 Sistem Yang Diusulkan

Tabel 4.1 Penjelasan *Use Case* Diagram

Nama Use Case	Deskripsi Use Case
Aktor (Pengguna)	Pengguna yang akan menjalankan aplikasi akan menampilkan sebuah Halaman Awal di mana terdapat beberapa pilihan menu.
Deteksi Kue Tradisional Bugis	Pengguna mengambil gambar kue tradisional bugis menggunakan kamera dan program akan menampilkan hasil deteksinya.
Hasil Deteksi Kue Tradisional Bugis	Berisi tentang nama kue dan Tingkat keyakinan model berdasarkan hasil deteksi gambar yang telah diinputkan sebelumnya.
Katalog Kue Tradisional Bugis	Berisi tentang katalog kue tradisional bugis, serta menu yang dapat dipilih untuk melihat penjelasan tentang kue tradisional tersebut.
Halaman Deskripsi Kue Tradisional Bugis	Berisi tentang deskripsi singkat tentang kue tradisional bugis terkait Penjelasan, Resep, Cara Membuat, dan Referensi.
Tentang Aplikasi	Berisi tentang data penjelasan singkat tentang aplikasi terkait tujuan aplikasi dan teknologi yang digunakan
Keluar	Menutup aplikasi

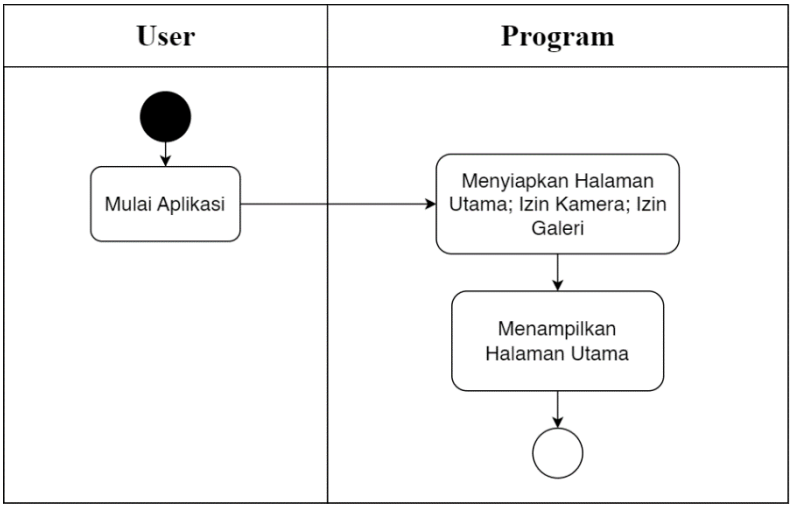
3. *Activity Diagram*

a. *Activity Diagram* Aplikasi



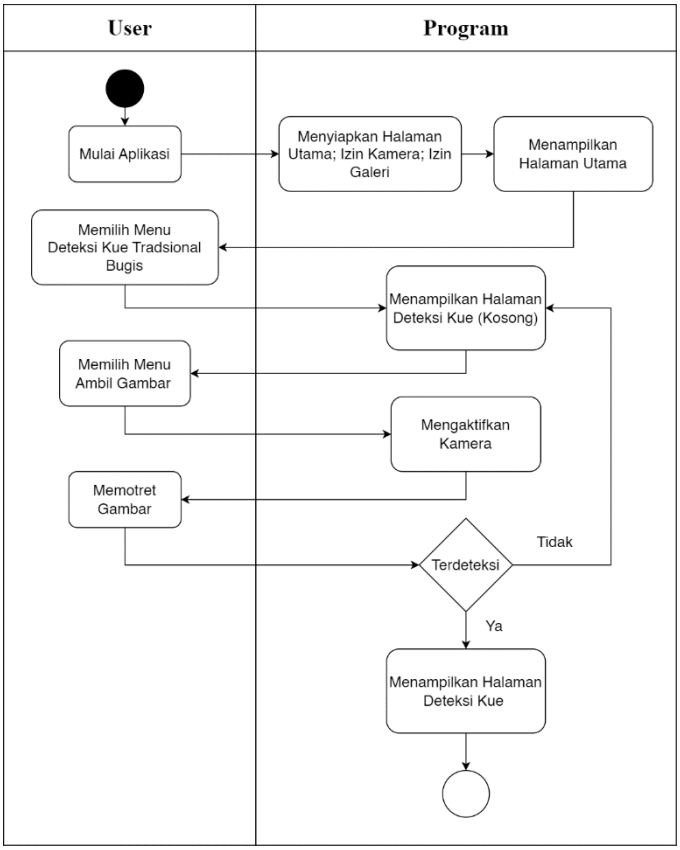
Gambar 4.3 *Activity Diagram* Aplikasi

b. *Activity Diagram* Mulai



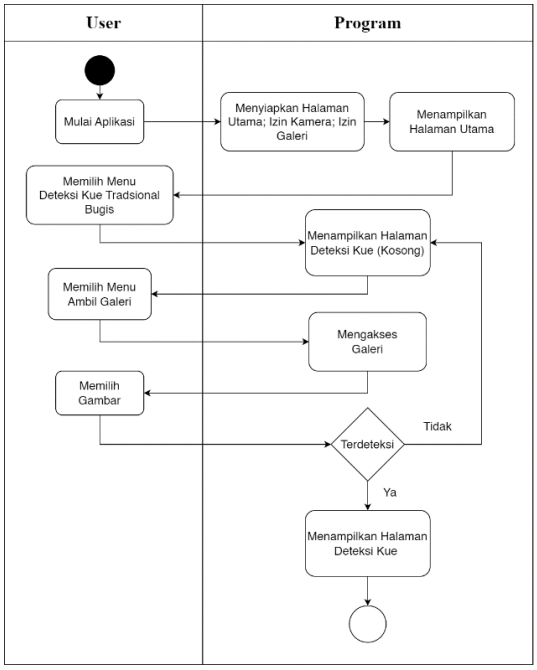
Gambar 4.4 *Activity Diagram* Mulai

c. *Activity Diagram* Deteksi Kue Tradisional Bugis Menggunakan Kamera



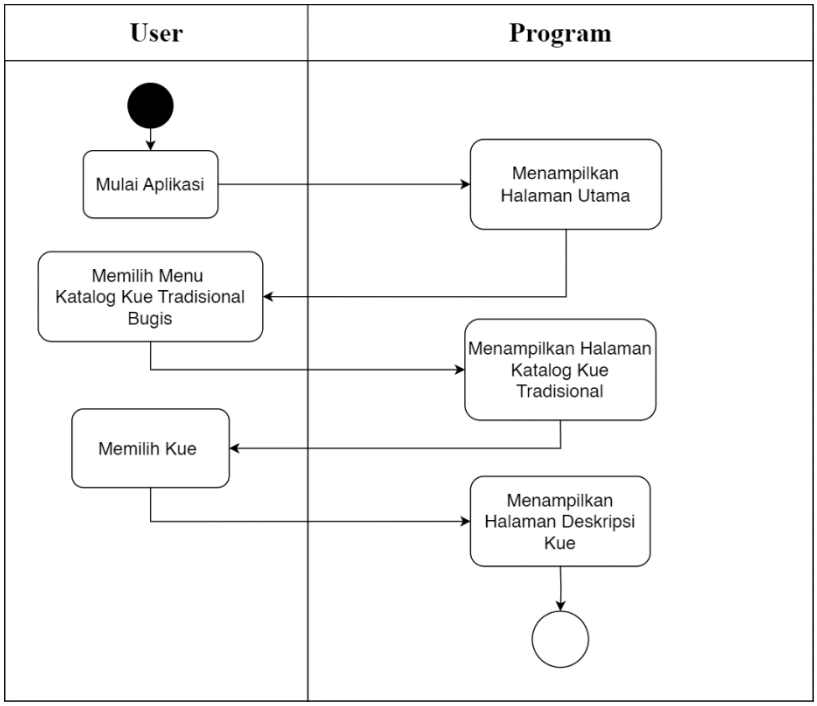
Gambar 4.5 *Activity Diagram* Deteksi Kue Tradisional Bugis menggunakan Kamera

d. Activity Diagram Deteksi Kue Tradisional Bugis menggunakan Galeri



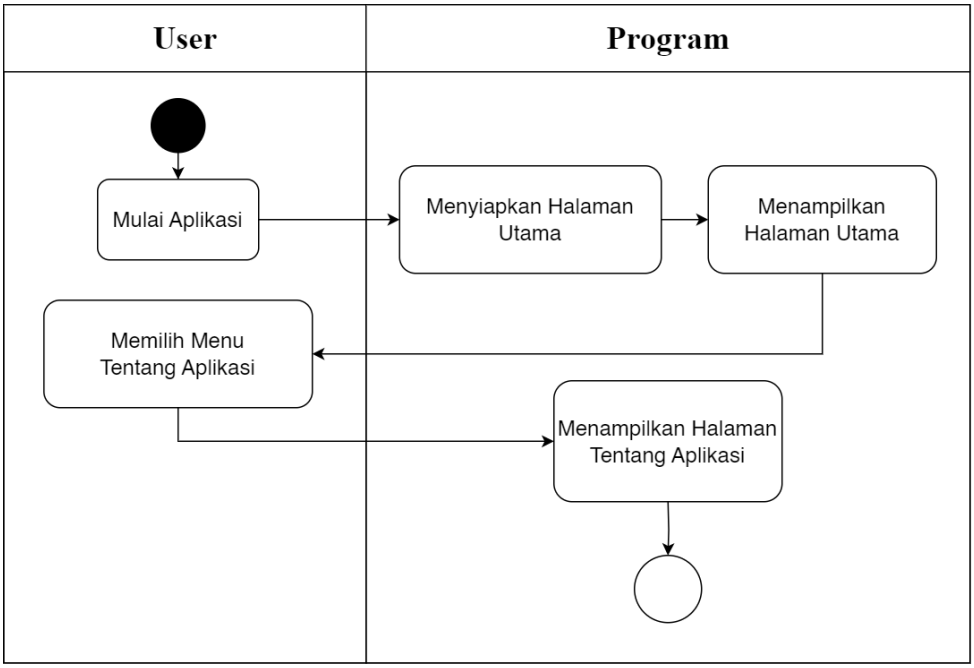
Gambar 4.6 *Activity Diagram* Deteksi Kue menggunakan Galeri

e. *Activity Diagram* Katalog Kue Tradisional Bugis



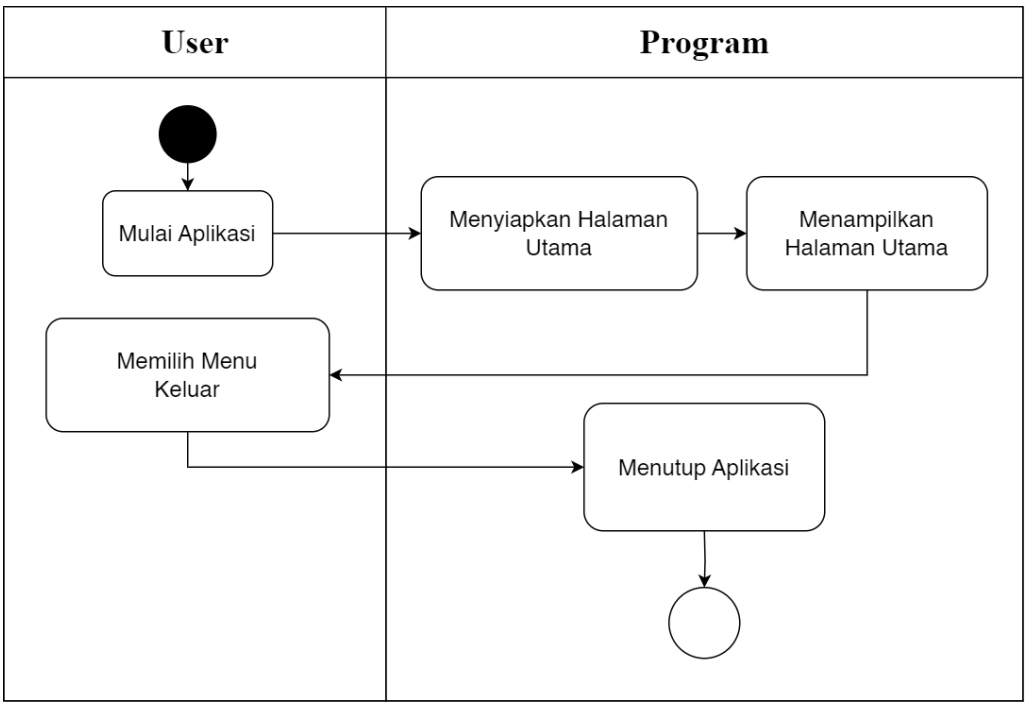
Gambar 4.7 *Activity Diagram* Katalog Kue Tradisional Bugis

f. *Activity Diagram* Tentang Aplikasi



Gambar 4.8 *Activity Diagram About*

g. *Activity Diagram* Keluar



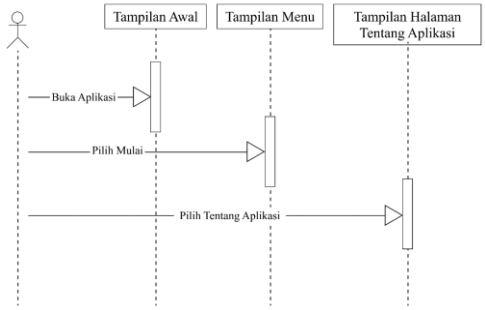
Gambar 4.9 *Activity Diagram* Keluar

Penjelasan terkait Gambar *Activity Diagram* di atas adalah sebagai berikut: ketika aplikasi dijalankan, pengguna akan disajikan menu utama yang terdiri dari beberapa opsi, yaitu "Deteksi Kue Tradisional", "Katalog Kue Tradisional Bugis", "Tentang Aplikasi", dan "Keluar". Apabila pengguna memilih opsi "Deteksi Kue Tradisional", aplikasi akan mengarahkan pengguna ke halaman Deteksi Kue, yang menyediakan dua opsi untuk mendeteksi kue, yaitu "Ambil Gambar" dan "Pilih dari Galeri". Saat pengguna memilih "Ambil Gambar", aplikasi akan mengakses kamera untuk mengambil gambar kue tradisional Bugis. Jika proses pengambilan gambar tidak berhasil, pengguna disarankan untuk mencoba kembali hingga gambar berhasil terdeteksi. Setelah gambar berhasil diambil dan terdeteksi, aplikasi akan menampilkan hasil deteksi pada halaman Deteksi Kue. Pengguna kemudian dapat kembali ke menu utama dan memilih opsi "Keluar".

4. *Sequence* Diagram

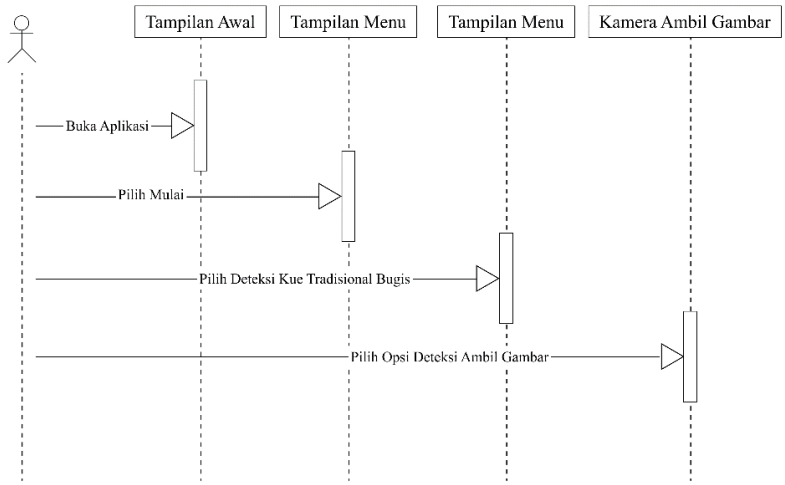
Diagram urutan (*Sequence Diagram*) ialah sejenis diagram interaksi yang memaparkan alir pelaksanaan sebuah tindakan, disertai dengan pesan yang dikirimkan serta waktu terjadinya setiap langkah.

a. *Sequence Diagram* Tampilan Awal



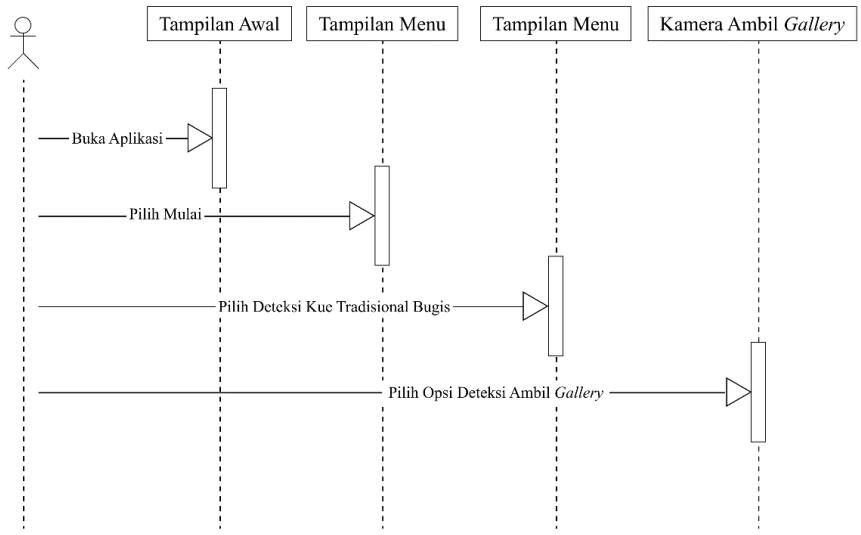
Gambar 4.10 *Sequence Diagram* Tampilan Aplikasi

b. *Sequence Diagram* Deteksi Kue Tradisional menggunakan Kamera



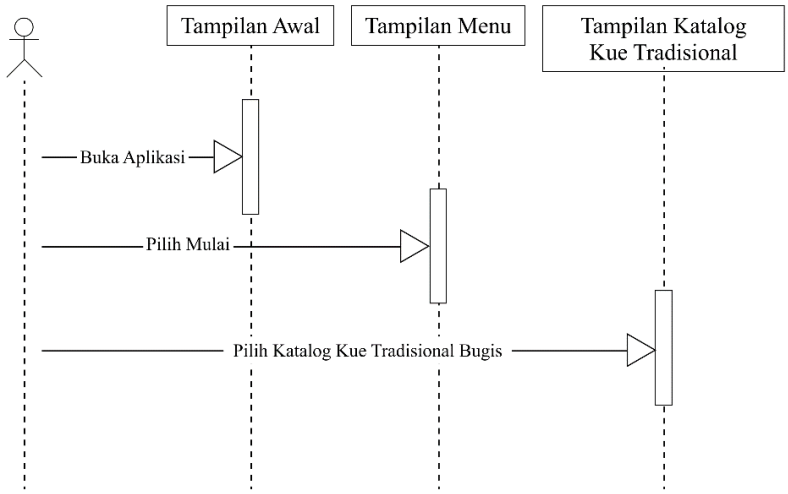
Gambar 4.11 *Sequence Diagram* Deteksi Kue Tradisional Bugis menggunakan Kamera

c. *Sequence Diagram* Deteksi Kue Tradisional menggunakan Galeri



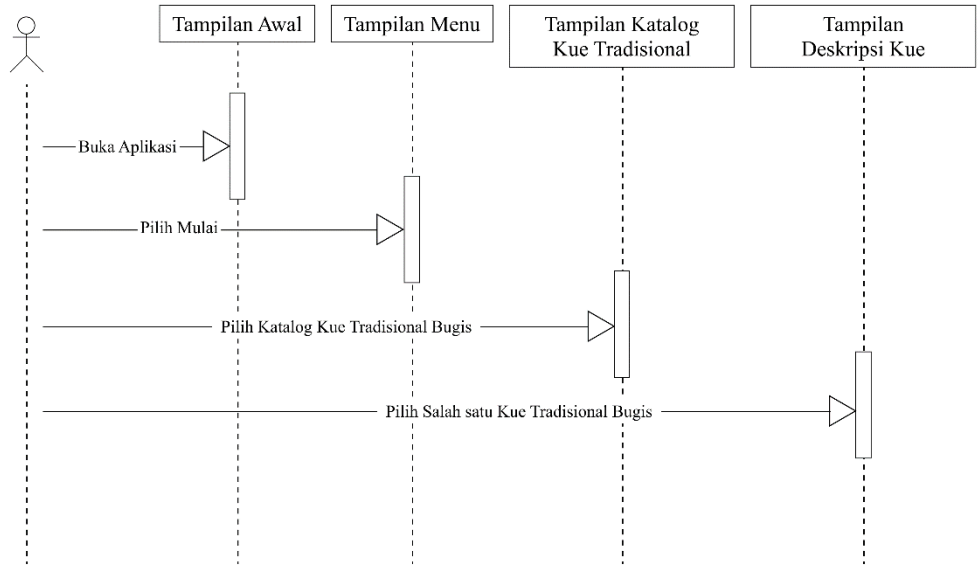
Gambar 4.12 *Sequence Diagram* Deteksi Kue Tradisional Bugis menggunakan Galeri

d. *Sequence Diagram* Katalog Kue Tradisional Bugis



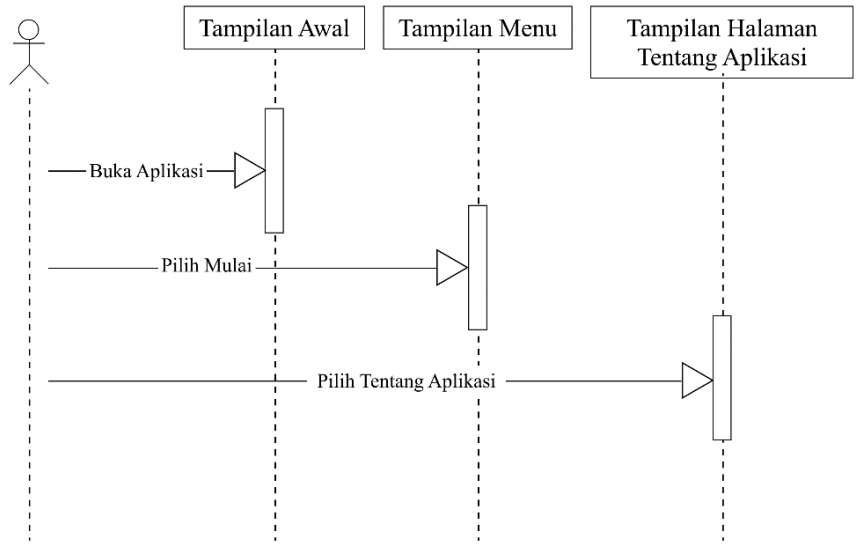
Gambar 4.13 *Sequence Diagram* Katalog Kue Tradisional Bugis

e. *Sequence Diagram* Halaman Deskripsi Kue



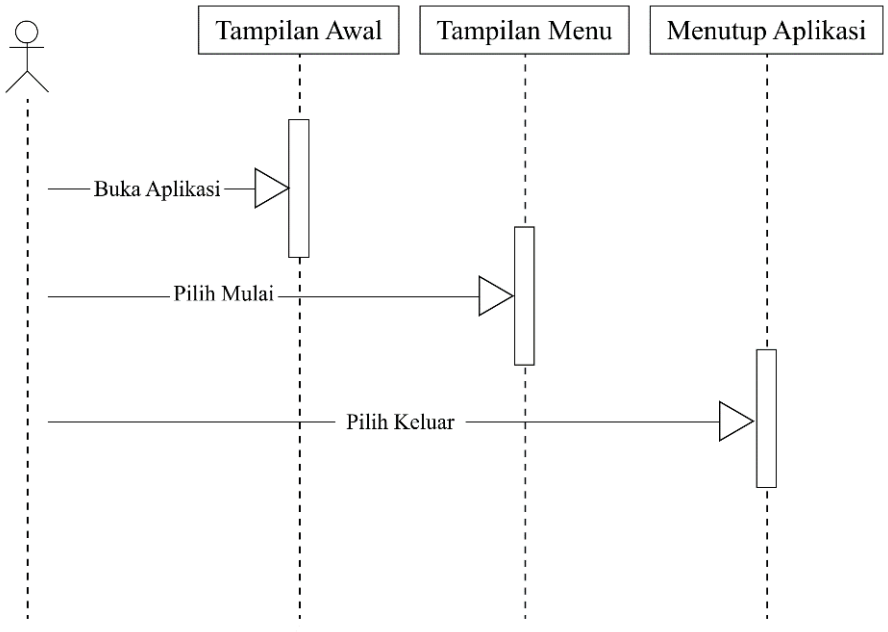
Gambar 4.14 *Sequence Diagram* Halaman Deskripsi Kue

f. *Sequence Diagram* Tentang Aplikasi



Gambar 4.15 *Sequence Diagram* Tentang Aplikasi

g. *Sequence Diagram* Keluar



Gambar 4.16 *Sequence Diagram* Keluar

B. Pengumpulan Data

Pengumpulan data merupakan langkah awal dan salah satu hal paling dasar dalam proses pengembangan model pembelajaran mesin. Dalam hal ini pengenalan gambar kue tradisonal Bugis menggunakan CNN, dataset yang digunakan terdiri dari 5 kategori kue tradisional bugis yaitu kue barongko, kue lapis, kue dange, kue taripang, dan kue onde-onde. Data ini diperoleh dari berbagai sumber melalui internet dan dokumentasi penulis. Data yang diperoleh sebanyak 2175 gambar, kemudian data dibagi menjadi 3 bagian yaitu, data latih yang berisi sebanyak 1175 gambar, data validasi yang berisi sebanyak 500 gambar, dan data testing yang berisi sebanyak 500 gambar, dan gambar lalu kemudian dimasukkan ke dalam Google Drive untuk dapat diakses nantinya.

C. Pembuatan Model CNN

Dalam pembuatan model CNN, penulis menggunakan google colab dalam menjalakan proses pemodelan ini. Hal pertama yang dilakukan untuk menjalankan pemodelan menggunakan google colab yaitu menyambung pada google drive untuk mengakses dataset yang telah dikumpulkan.

```
from google.colab import drive
drive.mount('content/drive');
```

Sumber kode di atas bertujuan mempertautkan yang ada pada Google Drive ke Google Colab. Setelah perintah dijalankan, sistem akan membuat *Uniform Resource Locator* (URL) autentikasi Google yang memungkinkan akun terhubung ke Google Colab. Setelah terkoneksi berhasil, langkah berikutnya meng-*import library* yang digunakan untuk memfasilitasi kelancaran pelaksanaan perintah dan

mendukung proses pembuatan dan pelatihan model.. Di bawah ini *Source code* untuk mengimport *library* yang diperlukan.

```
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers, models #type: ignore
from tensorflow.keras.preprocessing.image import
ImageDataGenerator #type: ignore
from tensorflow.keras.applications import MobileNetV2,
MobileNetV3Small, EfficientNetB0#type: ignore
from tensorflow.keras.applications.mobilenet_v2 import
preprocess_input as preprocessor_mobilenetv2 #type: ignore
from tensorflow.keras.applications.efficientnet import
preprocess_input as preprocessor_efficientnet #type: ignore
from tensorflow.keras.applications.mobilenet_v3 import
preprocess_input as preprocessor_mobilenetv3 #type: ignore
from tensorflow.keras.optimizers import Adam #type: ignore
from tensorflow.keras.callbacks import ReduceLROnPlateau,
EarlyStopping #type: ignore
from tensorflow.keras.layers import Dense,
GlobalAveragePooling2D, Dropout, Concatenate, Input, Average
#type: ignore
from tensorflow.keras.models import Model #type: ignore
from tensorflow.keras.utils import plot_model #type: ignore
from sklearn.metrics import confusion_matrix,
classification_report, precision_score, recall_score, f1_score,
accuracy_score
from collections import Counter
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import os
import json
```

D. *Preprocessing*

Setelah dataset kue berhasil dikumpulkan, langkah selanjutnya adalah tahap *preprocessing*. Semua data tersebut akan diubah ukurannya atau diresize untuk menyamakan ukuran gambar tersebut. Semua ukuran dataset gambar akan ditransformasi (*resize*) menjadi 224 x 224 pixel untuk memastikan keseragaman ukuran. Setelah itu, setiap data diberi label berdasarkan nama folder yang ada pada *Google Drive*.

```
data_dir = '/content/drive/MyDrive/image-dataset/kue/DATASET
v3.2.1'
total_images = sum([len(files) for _, _, files in
os.walk(data_dir)])
print(f"Jumlah data di dataset\t\t: {total_images} gambar")
print('-------------------------------------------------')

train_dir = data_dir + '/training'
val_dir = data_dir + '/validation'
test_dir = data_dir + '/testing'

def count_files(directory):
    total_files = 0
    for subdir, _, files in os.walk(directory):
        total_files += len(files)
    return total_files

# Menampilkan jumlah file dalam setiap direktori
print(f"Jumlah data di training set\t: {count_files(train_dir)}
gambar ")
print(f"Jumlah data di validation set\t: {count_files(val_dir)}
gambar")
print(f"Jumlah data di testing set \t: {count_files(test_dir)}
gambar")
```

Berdasarkan *source code* di atas mengambil dataset dari folder kue yang terbagi atas folder *train* dan folder *test*. Selanjutnya dimensi ukuran seluruh gambar di*resize* menjadi 224 x 224 pixel. Setelah ini dilakukan proses yang dikenal juga sebagai data augmentation. Data augmentation adalah teknik dalam pemodelan CNN yang bertujuan untuk meningkatkan variasi data tanpa perlu mengumpulkan data tambahan. Teknik ini membantu model menjadi lebih robust terhadap berbagai kondisi nyata dengan menerapkan transformasi seperti rotasi, pergeseran, zoom, flipping, serta perubahan warna atau kontras pada gambar. Dengan cara ini, model dapat belajar dari variasi data yang lebih luas dan menghindari overfitting.

Untuk tahap pelatihan model, dilakukan proses augmentasi dan normalisasi data menggunakan objek ImageDataGenerator dari Keras. Potongan kode berikut merupakan salah satu contoh implementasi generator data yang digunakan dalam penelitian ini, dan khususnya digunakan dalam arsitektur MobileNetV3Small.

Namun, pendekatan yang sama juga diterapkan pada arsitektur lainnya, yaitu MobileNetV2 dan EfficientNetB0.

```
def create_generators(batch_size, seed=42):
    train_datagen = ImageDataGenerator(
        preprocessing_function=preprocessor_mobilenetv3,
        rotation_range=15,
        width_shift_range=0.2,
        height_shift_range=0.2,
        shear_range=0.1,
        zoom_range=0.2,
        horizontal_flip=True,
        brightness_range=(0.9, 1.1),
        fill_mode='nearest',
    )
    val_datagen = ImageDataGenerator(
        preprocessing_function=preprocessor_mobilenetv3,
    )
    test_datagen = ImageDataGenerator(
        preprocessing_function=preprocessor_mobilenetv3,
    )

    train_gen = train_datagen.flow_from_directory(
        train_dir,
        target_size=(224, 224),
        batch_size=batch_size,
        class_mode="categorical",
        shuffle=True,
        seed=seed
    )
    val_gen = val_datagen.flow_from_directory(
        val_dir,
        target_size=(224, 224),
        batch_size=batch_size,
        class_mode="categorical",
        shuffle=False
    )
    test_gen = test_datagen.flow_from_directory(
        test_dir,
        target_size=(224, 224),
        batch_size=batch_size,
        class_mode="categorical",
        shuffle=False
    )
    return train_gen, val_gen, test_gen
print("==== BATCH SIZE 16 ====")
train_gen16, val_gen16, test_gen16 =
create_generators(batch_size=16, seed=42)
print("==== BATCH SIZE 32 ====")
train_gen32, val_gen32, test_gen32 =
create_generators(batch_size=32, seed=42)
print("==== BATCH SIZE 64 ====")
train_gen64, val_gen64, test_gen64 =
create_generators(batch_size=64, seed=42)
```

Dalam kode di atas, proses normalisasi dilakukan melalui parameter preprocessing_function, yang menggunakan fungsi praproses khusus untuk arsitektur MobileNetV3 (preprocessor_mobilenetv3). Fungsi ini menyesuaikan skala nilai piksel gambar agar sesuai dengan kebutuhan model pretrained MobileNetV3Small. Normalisasi ini penting untuk meningkatkan efektivitas dan stabilitas proses pembelajaran dengan memastikan bahwa data input berada dalam rentang nilai yang sesuai dengan bobot awal dari model pretrained.

Selain normalisasi, dilakukan juga berbagai teknik augmentasi data pada data pelatihan untuk meningkatkan keberagaman data dan mencegah *overfitting*. Rotation range digunakan untuk memutar gambar secara acak dalam rentang 15 derajat untuk membatu model menjadi lebih robust terhadap rotasi gambar, width dan height shift range digunakan untuk menggeser gambar secara horizontal dan vertical sebesar 20% dari dimensi gambar. Ini menambah variasi data dengan mensimulasikan kondisi pengambilan gambar yang sedikit bergeser. Shear berfungsi untuk menerapkan transformasi shaering sebesar 10% untuk menghasilkan distorsi geometris. Zoom sebesar 0.2 akan memperbesar atau memperkecil gambar hingga 20% untuk memberikan variasi skala objek. Fill mode berfungsi untuk mengisi piksel kosong yang mungkin terjadi setelah transformasi dengan nilai piksel terdekat, memastikan tidak ada area kosong di gambar yang dapat mengganggu pelatihan model. Brightness range sebesar 0.9 dan 1.1 akan menyesuaikan kecerahan gambar dalam kisaran 90% hingga 110%, yang membantu model beradaptasi dengan variasi pencahayaan. Horizontal flip berfungsi untuk membalik gambar secara horizontal yang membantu meningkatkan

variasi data dengan menciptakan perspektif berbeda dari objek dalam gambar. Teknik-teknik ini berguna untuk membuat model lebih robust terhadap perubahan orientasi, terutama dalam kasus di mana orientasi objek tidak tetap atau bisa muncul dalam berbagai sudut pandang.

E. Modelling (Perancangan Model)

Pada tahap perancangan model, setelah menyamakan seluruh ukuran gambar dan telah memberikan label akan melalui proses pemodelan menggunakan algoritma atau metode CNN. Arsitektur yang diuji dalam perancangan model ini adalah MobileNetV2, MobileNetV3Small, EfficientNetB0, yang merupakan arsitektur-arsitektur *Convolutional Neural Network* (CNN) yang banyak digunakan dalam pengolahan gambar yang dirancang khusus untuk kebutuhan komputasi yang lebih efisien atau dengan sumber daya terbatas.

Agar model dapat beradaptasi dengan data yang baru, dilakukan penambahan beberapa layer di atas arsitektur bawaan (pretrained base model), yaitu layer GlobalAveragePooling2D, Dense layer dengan aktivasi ReLU untuk membentuk representasi fitur baru, serta Dropout layer untuk mengurangi risiko overfitting. Pada akhir arsitektur, ditambahkan layer Dense output dengan fungsi aktivasi softmax yang menyesuaikan jumlah kelas pada dataset

Dalam implementasinya, dilakukan pendekatan *transfer learning*, yaitu memanfaatkan model yang telah dilatih sebelumnya pada dataset besar seperti ImageNet, kemudian disesuaikan (*fine-tuned*) untuk menyelesaikan tugas klasifikasi pada dataset yang digunakan dalam penelitian ini. Salah satu contoh

implementasi transfer learning menggunakan arsitektur MobileNetV2 ditunjukkan pada potongan kode berikut:

```
base_model = MobileNetV2(input_shape=(224, 224, 3),
                         include_top=False,
                         weights='imagenet',
                         )

base_model.trainable = False

x = base_model.output
x = GlobalAveragePooling2D()(x)
x = Dense(32, activation='relu')(x)
x = Dropout(0.2)(x)
outputs = layers.Dense(num_classes, activation="softmax")(x)

MobileNetV2_model = keras.Model(
    inputs=base_model.input,
    outputs=outputs,
    name="MobileNetV2_epoch10_batch16"
)

MobileNetV2_model.compile(
    optimizer=Adam(learning_rate=0.0001),
    loss="categorical_crossentropy",
    metrics=["accuracy"]
    )

MobileNetV2_model.summary()
```

Kode di atas hanya merupakan salah satu contoh implementasi dari pemodelan yang digunakan. Pendekatan serupa juga diterapkan pada arsitektur MobileNetV3Small dan EfficientNetB0 untuk dibandingkan performanya. Penggunaan *transfer learning* dalam hal ini memungkinkan model untuk belajar dari fitur-fitur yang telah dipelajari sebelumnya, sehingga dapat meningkatkan akurasi dan efisiensi pelatihan, terutama ketika dataset yang digunakan tidak terlalu besar.

F. Pelatihan dan Pengujian Model

Pada tahapan ini, model CNN dilatih menggunakan dataset yang telah dikumpulkan sebelumnya. Penggunaan generator data dari ImageDataGenerator

memungkinkan pelatihan dengan data yang ditingkatkan secara dinamis, meningatkan kemampuan generalisasi model dengan memperkenalkan variasi ke dalam dataset pelatihan melalu transformasi gambar seperti rotasi, pemangkasan, pembalikan, pengecilan, dan pembesaran.

Pada tahapan pelatihan ini, variasi nilai parameter dilakukan untuk perbandingan, parameter yang diinisialisasi meliputi ukuran *batch size* dan *epoch*. Inisialisasi ini lebih dari sekadar perbandingan dan dapat memberikan akurasi optimal dan kecepatan belajar yang tidak diketahui. Ukuran *batch size* yang diuji pada pelatihan ini adalah 16, 32, dan 64, sementara jumlah *epoch* yang diterapkan untuk pelatihan model adalah 10, 30, dan 50. Oleh sebab itu, penelitian ini dilakukan beberapa eksperimen untuk memperoleh nilai akurasi yang terbaik.

Untuk setiap arsitektur, dilakukan pelatihan menggunakan kombinasi parameter tersebut. Hasil dari proses pelatihan, termasuk metrik akurasi dan loss pada data pelatihan maupun validasi, kemudian dianalisis dan disajikan dalam bentuk tabel pada subbagian berikut. Adapun rincian proses pelatihan masing-masing arsitektur dibahas pada bagian berikut:

1. Pelatihan Model MobileNetV2

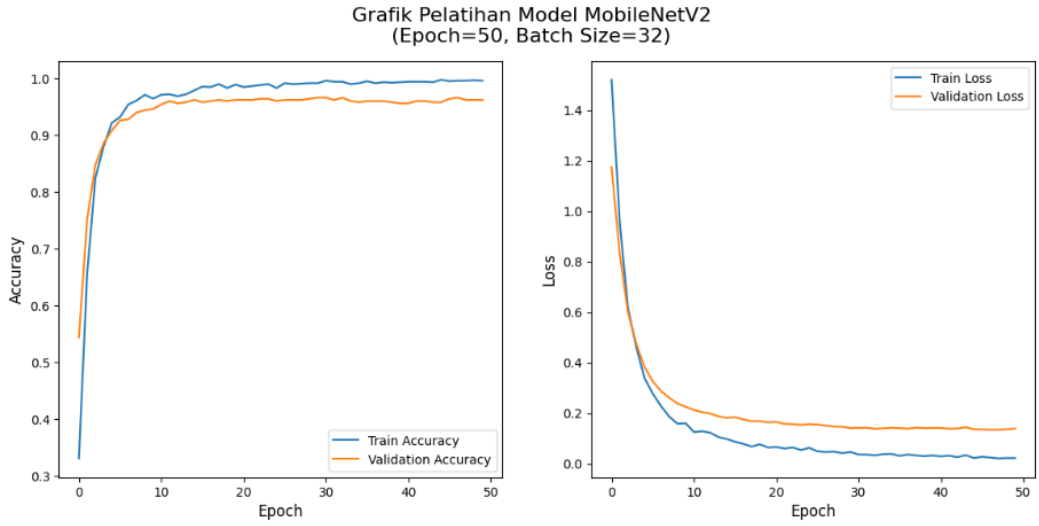
Tabel 4.2 Tabel Pelatihan Model MobileNetV2

No	Batch Size	Epoch	Accuracy	Loss	Validasi Accuracy	Validasi Loss	Test Accuracy	Test Loss
1	16	10	0.9813	0.0938	0.9520	0.1893	0.8940	0.3391
2	16	30	0.9983	0.0211	0.9500	0.1382	0.8980	0.3041
3	16	50	0.9991	0.0108	0.9460	0.1648	0.8920	0.3210
4	32	10	0.9643	0.1428	0.9420	0.2151	0.8620	0.4014

Lanjutan Tabel 4.2

No	Batch Size	Epoch	Accuracy	Loss	Validasi Accuracy	Validasi Loss	Test Accuracy	Test Loss
5	32	30	0.9872	0.0487	0.9540	0.1434	0.8920	0.3192
6	32	50	0.9957	0.0223	0.9620	0.1392	0.9060	0.2737
7	64	10	0.9515	0.2198	0.9280	0.3024	0.7940	0.6126
8	64	30	0.9804	0.0703	0.9500	0.1514	0.8600	0.3810
9	64	50	0.9906	0.0455	0.9600	0.1483	0.8780	0.3664

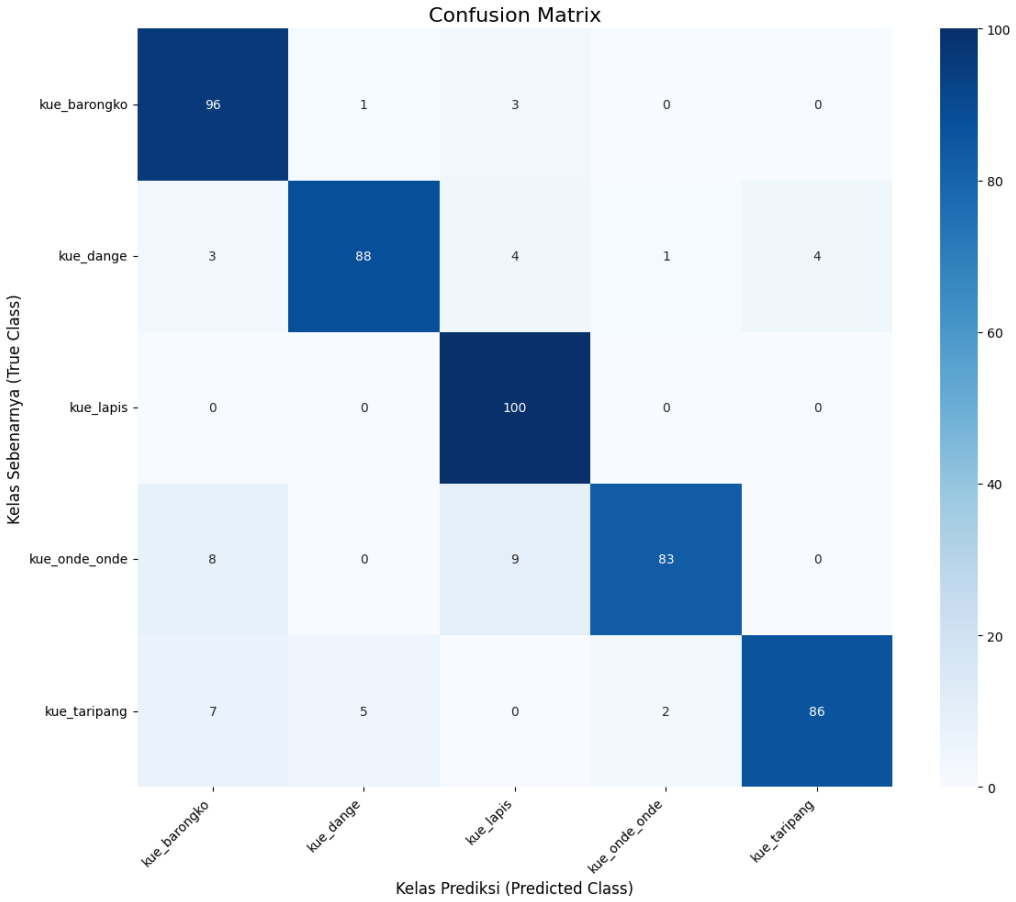
Berdasarkan Tabel 4.2, pelatihan dengan batch size 32 dan epoch 50 menghasilkan akurasi terbaik pada model yang dikembangkan. Akurasi yang tercapai pada model ini adalah 0.9957 untuk akurasi pelatihan, 0.9620 untuk akurasi validasi, dan 0.9060 untuk akurasi testing. Grafik akurasi serta loss dari model ini bisa diamati di Gambar 4.17.



Gambar 4.17 Hasil Grafik Akurasi dan Loss Model MobileNetV2

Pengujian dalam penelitian ini dilakukan menggunakan metode *Confusion Matrix*, yang merupakan salah satu teknik evaluasi kinerja dari suatu model klasifikasi dalam *machine learning*. *Confusion matrix* menampilkan representasi terkait seberapa baik atau seberapa buruk model dalam melakukan klasifikasi pada

dataset yang diberikan. Metode ini berguna untuk menghitung kllasifikasi, metode yang sangat berguna untuk mengutus recall, precision, akurasi, dan F1-score serta memberikan informasi jumlah kelas prediksi yang diberikan. Evaluasi dilakukan berdasarkan model terbaik yang diperoleh dari proses pelatihan. Model yang diuji dalam penelitian ini adalah hasil pelatihan dengan menggunakan *batch size* 32 dan *epoch* 50, yang menghasilkan *training accuracy* sebesar 0.9957, *validation accuracy* sebesar 0.9620, dan *testing accuracy* sebesar 0.9060. Hasil pengujian penelitian berdasarkan model terbaik dapat dilihat pada Gambar 4.18 berikut:



Gambar 4.18 Evaluasi Confusion Matrix MobileNetV2

Pada Gambar 4.18 di atas, dilakukan pengujian menggunakan *Confusion Matrix* untuk mengevaluasi kemampuan model dalam mengenali gambar berdasarkan dataset uji yang telah disediakan dengan menerapkan metode CNN. Hasil pengujian menunjukkan nilai *accuracy, recall, precision*, dan *F1-score* sebagai berikut:

Parameter evaluasi ini dihitung berdasarkan jumlah *True Positive (TP), True Negative (TN), False Negative (FN), dan False Positive (FP)* untuk masing-masing kelas. Berikut hasil perhitungan dari Akurasi, *Precision, Recall* dan *F1-Score* berdasarkan setiap kelasnya:

a. Kelas Kue Barongko

   TP = 96, FP = 18, FN = 4, dan TN = 382

b. Kelas Kue Dange

   TP = 88, FP = 6, FN = 12, dan TN = 394

c. Kelas Kue Lapis

   TP = 100, FP = 16, FN = 0, dan TN = 384

d. Kelas Kue Onde-Onde

   TP = 83, FP = 3, FN = 17, dan TN = 397

e. Kelas Kue Taripang

   TP = 86, FP = 4, FN = 14, dan TN = 396

   Berikut dibawah ini rumus dan hasil dari Akurasi, Precision, Recall, dan F1 Score:

a. Akurasi

   Akurasi bisa diperoleh memakai formula

$$Akurasi = \frac{\text{Jumlah Prediksi Benar}}{\text{Total Sampel}} \quad \text{..... (1)}$$

Total data uji = 500

Total Prediksi Benar = 96 + 88 + 100 + 83 + 86 = 453

Akurasi = $\frac{453}{500} = 0.906\ atau\ 90.6\%$

b. Precision

Precision dapat dihitung dengan rumus berikut:

$$Precision = \frac{\text{TP}}{\text{TP+FP}} \quad \text{..... (2)}$$

1) Kue Barongko, Precision = $\frac{96}{96+18} = 0.8421$

2) Kue Dange, Precision = $\frac{88}{88+6} = 0.9362$

3) Kue Lapis, Precision = $\frac{100}{100+16} = 0.8621$

4) Kue Onde-Onde, Precision = $\frac{83}{83+3} = 0.9651$

5) Kue Taripang, Precision = $\frac{86}{86+4} = 0.9556$

c. Recall

Recall dapat dihitung menggunakan rumus berikut:

$$Recall = \frac{\text{TP}}{\text{TP+FN}} \quad \text{..... (3)}$$

1) Kue Barongko, Recall = $\frac{96}{96+4} = 0.8421$

2) Kue Dange, Recall = $\frac{88}{88+12} = 0.88$

3) Kue Lapis, Recall = $\frac{100}{100+0} = 1$

4) Kue Onde-Onde, Recall = $\frac{83}{83+17} = 0.83$

5) Kue Taripang, Recall = $\frac{86}{86+14} = 0.86$

d. F1-Score

F1-Score dapat dihitung menggunakan rumus berikut:

$$F1 - Score = 2\ x\ \frac{\text{Precision x Recall}}{\text{Precision+Recall}} \quad \text{..... (4)}$$

1) Kue Barongko, F1-Score $= 2\ x\ \frac{0.8421 \times 0.96}{0.8421+0.96} = 0.8972$

2) Kue Dange, F1-Score $= 2\ x\ \frac{0.9362 \times 0.88}{0.9362+0.88} = 0.9072$

3) Kue Lapis, F1-Score $= 2\ x\ \frac{0.8621 \times 1.00}{0.8621+1.00} = 0.9254$

4) Kue Onde-Onde, F1-Score $= 2\ x\ \frac{0.9651 \times 0.83}{0.9651+0.83} = 0.8926$

5) Kue Taripang, F1-Score $= 2\ x\ \frac{0.9556 \times 0.86}{0.9556+0.86} = 0.9053$

Berdasarkan evaluasi menggunakan Confusion Matrix, model MobileNetV2 menunjukkan performa yang sangat memuaskan dengan akurasi keseluruhan mencapai 90.6%. Akurasi yang tinggi ini menandakan bahwa model secara umum berhasil mempelajari fitur-fitur visual yang membedakan kelima kelas kue secara efektif. Nilai F1-Score yang konsisten tinggi di semua kelas (berkisar antara 0.89 hingga 0.92) juga mengonfirmasi bahwa model memiliki keseimbangan yang baik antara presisi dan recall, menjadikannya andal dan robust.

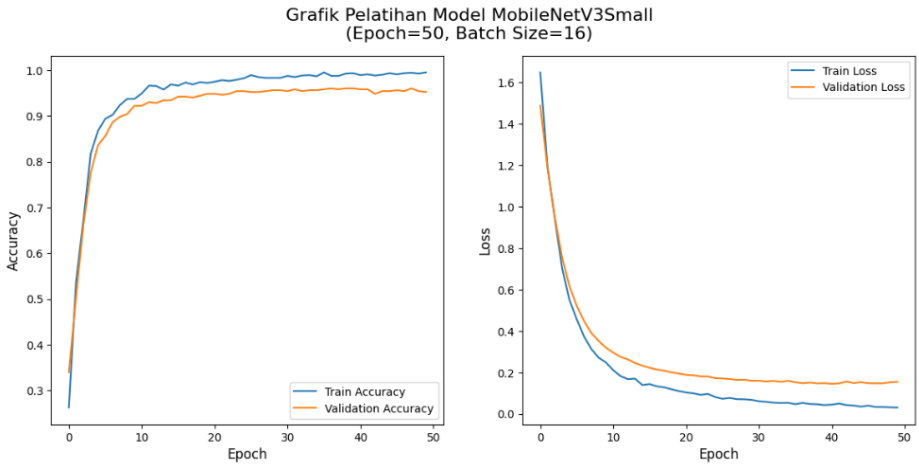
Analisis lebih dalam pada setiap kelas menunjukkan karakteristik performa yang menarik. Kue Lapis menjadi kelas yang paling mudah dikenali oleh model, terbukti dengan nilai Recall sempurna (1.0), yang berarti tidak ada satupun gambar Kue Lapis yang terlewatkan. Di sisi lain, Kue Onde-Onde memiliki Precision tertinggi (0.9651), yang artinya prediksi model untuk kelas ini memiliki tingkat kebenaran yang sangat tinggi.

2. Pelatihan Model MobileNetV3Small

Tabel 4.3 Tabel Pelatihan Model MobileNetV3Small

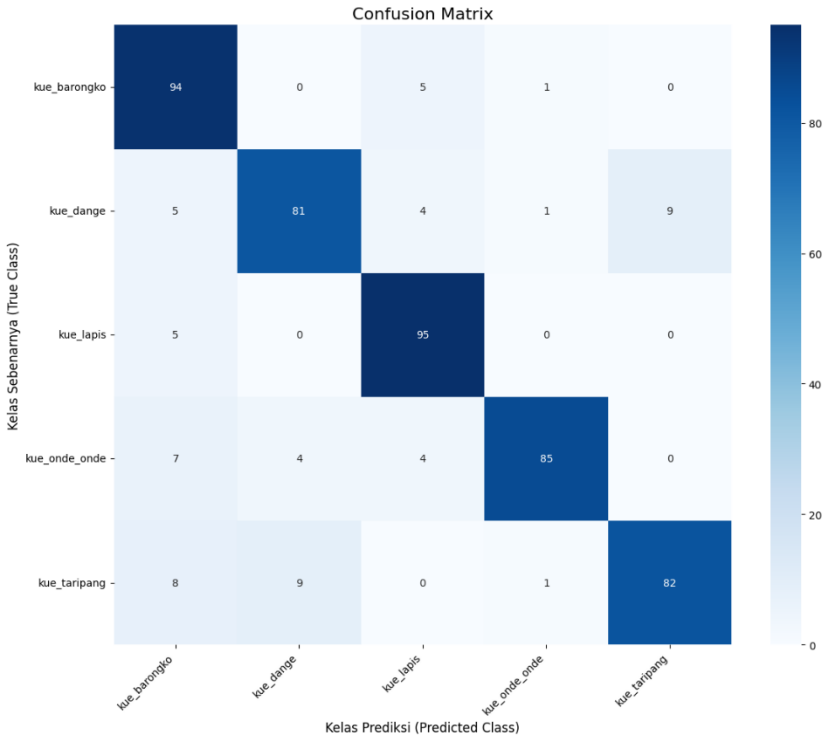
No	Batch Size	Epoch	Accuracy	Loss	Validasi Accuracy	Validasi Loss	Test Accuracy	Test Loss
1	16	10	0.9821	0.0719	0.9560	0.1260	0.9000	0.3565
2	16	30	0.9864	0.0658	0.9560	0.1492	0.8400	0.4307
3	16	50	0.9949	0.0306	0.9520	0.1548	0.8740	0.3608
4	32	10	0.9106	0.3845	0.8760	0.4379	0.7400	0.7703
5	32	30	0.9770	0.0957	0.9560	0.1502	0.8480	0.4005
6	32	50	0.9787	0.0614	0.9640	0.1546	0.8440	0.4319
7	64	10	0.8567	0.4733	0.8860	0.4605	0.7040	0.8671
8	64	30	0.9609	0.1605	0.9380	0.2176	0.8040	0.5492
9	64	50	0.9702	0.1113	0.9460	0.1748	0.8280	0.4482

Berdasarkan Tabel 4.3, pelatihan dengan batch size 16 dan epoch 50 menghasilkan akurasi terbaik pada model yang dikembangkan. Akurasi yang tercapai pada model ini adalah 0.9949 untuk akurasi pelatihan, 0.9520 untuk akurasi validasi, dan 0.8740 untuk akurasi testing. Grafik akurasi serta loss dari model ini bisa diamati di Gambar 4.19



Gambar 4.19 Hasil Grafik Akurasi dan Loss Model MobileNetV3 Small

Pengujian dalam penelitian ini dilakukan menggunakan metode *Confusion Matrix*, yang merupakan salah satu teknik evaluasi kinerja dari suatu model klasifikasi dalam *machine learning*. *Confusion matrix* menampilkan representasi terkait seberapa baik atau seberapa buruk model dalam melakukan klasifikasi pada dataset yang diberikan. Metode ini berguna untuk menghitung kllasifikasi, metode yang sangat berguna untuk mengutus recall, precision, akurasi, dan F1-score serta memberikan informasi jumlah kelas prediksi yang diberikan. Evaluasi dilakukan berdasarkan model terbaik yang diperoleh dari proses pelatihan. Model yang diuji dalam penelitian ini adalah hasil pelatihan dengan menggunakan *batch size* 16 dan *epoch* 50, yang menghasilkan *training accuracy* sebesar 0.9949, *validation accuracy* sebesar 0.9520, dan *testing accuracy* sebesar 0.8740. Hasil pengujian penelitian berdasarkan model terbaik dapat dilihat pada Gambar 4.20 berikut:



Gambar 4.20 Evaluasi Confusion Matrix MobileNetV3 Small

Pada Gambar 4.20, dilakukan pengujian menggunakan *Confusion Matrix* untuk mengevaluasi kemampuan model dalam mengenali gambar berdasarkan dataset uji yang telah disediakan dengan menerapkan metode CNN. Hasil pengujian menunjukkan nilai *accuracy, recall, precision*, dan *F1-score* sebagai berikut:

Parameter evaluasi ini dihitung berdasarkan jumlah *True Positive (TP), True Negative (TN), False Negative (FN), dan False Positive (FP)* untuk masing-masing kelas. Berikut hasil perhitungan dari Akurasi, *Precision, Recall* dan *F1-Score* berdasarkan setiap kelasnya:

a. Kelas Kue Barongko

   TP = 94, FP = 25, FN = 6, dan TN = 375

b. Kelas Kue Dange

   TP = 81, FP = 13, FN = 19, dan TN = 387

c. Kelas Kue Lapis

   TP = 95, FP = 13, FN = 5, dan TN = 387

d. Kelas Kue Onde-Onde

   TP = 85, FP = 3, FN = 15, dan TN = 397

e. Kelas Kue Taripang

   TP = 82, FP = 9, FN = 18, dan TN = 391

   Berikut dibawah ini rumus dan hasil dari Akurasi, Precision, Recall, dan F1 Score:

a. Akurasi

   Total data uji = 500

   Total Prediksi Benar = 94 + 81 + 95 + 85 + 82 = 437

Akurasi = $\frac{437}{500} = 0.874\ atau\ 87.4\%$

b. Precision

1) Kue Barongko, Precision = $\frac{94}{94+25} = 0.7899$

2) Kue Dange, Precision = $\frac{81}{81+13} = 0.8617$

3) Kue Lapis, Precision = $\frac{95}{95+13} = 0.8796$

4) Kue Onde-Onde, Precision = $\frac{85}{85+3} = 0.9659$

5) Kue Taripang, Precision = $\frac{82}{82+9} = 0.9011$

c. Recall

1) Kue Barongko, Recall = $\frac{94}{94+6} = 0.94$

2) Kue Dange, Recall = $\frac{81}{81+19} = 0.81$

3) Kue Lapis, Recall = $\frac{95}{95+5} = 0.95$

4) Kue Onde-Onde, Recall = $\frac{85}{85+15} = 0.85$

5) Kue Taripang, Recall = $\frac{82}{82+18} = 0.82$

d. F1-Score

1) Kue Barongko, F1-Score = $2\ x\ \frac{0.7899 \times 0.94}{0.7899+ 0.94} = 0.8585$

2) Kue Dange, F1-Score = $2\ x\ \frac{0.8617 \times 0.81}{0.8617+ 0.81} = 0.8352$

3) Kue Lapis, F1-Score = $2\ x\ \frac{0.8796 \times 0.95}{0.8796+ 0.95} = 0.9132$

4) Kue Onde-Onde, F1-Score = $2\ x\ \frac{0.9659 \times 0.85}{0.9659+ 0.85} = 0.9046$

5) Kue Taripang, F1-Score = $2\ x\ \frac{0.9011 \times 0.82}{0.9011+ 0.82} = 0.8587$

Berdasarkan evaluasi Confusion Matrix, model yang diuji menunjukkan kinerja yang kompeten dengan akurasi keseluruhan sebesar 87.4%. Hasil ini mengonfirmasi bahwa metode CNN yang digunakan mampu menangani tugas klasifikasi citra kue dengan baik. Performa terbaik dicatatkan oleh Kue Lapis, yang mencapai F1-Score tertinggi (0.9132), menandakan keseimbangan yang sangat baik antara presisi dan recall. Selain itu, Kue Onde-Onde menunjukkan Precision tertinggi (0.9659), menjadikan prediksinya untuk kelas ini sangat dapat diandalkan.

Meskipun demikian, terdapat beberapa tantangan. Kue Barongko memiliki Precision terendah (0.7899) karena tingginya angka False Positive, yang berarti model cenderung sering salah menebak kelas ini. Sementara itu, Kue Dange menjadi kelas yang paling sulit dikenali, terbukti dari Recall terendah (0.81). Secara keseluruhan, model ini terbukti efektif, namun ada ruang untuk perbaikan, terutama dalam meningkatkan kemampuan model untuk membedakan kelas-kelas yang memiliki kemiripan visual.

3. Pelatihan Model EfficientNetB0

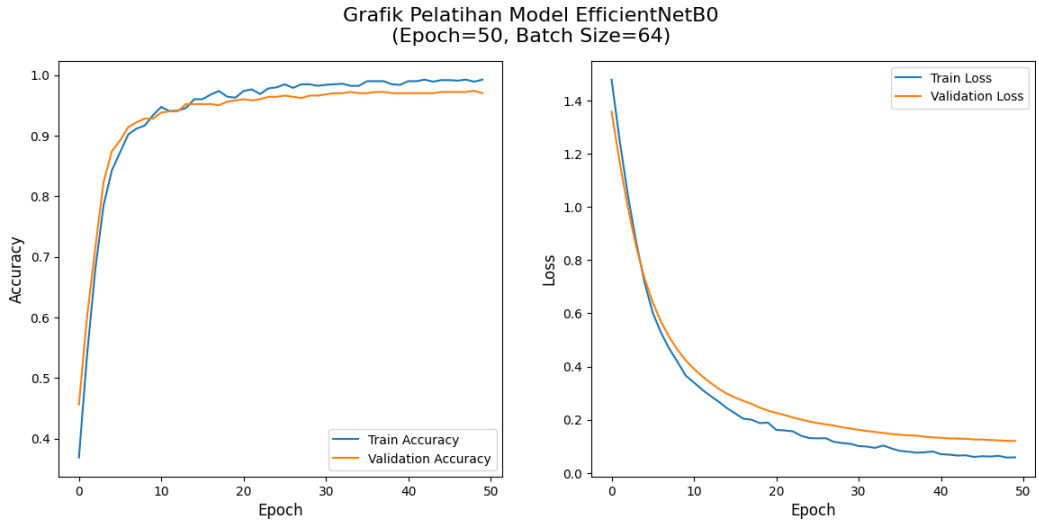
Tabel 4.4 Tabel Pelatihan Model EfficientNetB0

No	Batch Size	Epoch	Accuracy	Loss	Validasi Accuracy	Validasi Loss	Test Accuracy	Test Loss
1	16	10	0.9728	0.1233	0.9620	0.1701	0.8920	0.3588
2	16	30	0.9966	0.0259	0.9740	0.0979	0.9080	0.2721
3	16	50	0.9974	0.0183	0.9780	0.0823	0.9220	0.2169
4	32	10	0.9711	0.1947	0.9500	0.2636	0.8540	0.4517
5	32	30	0.9940	0.0471	0.9680	0.1128	0.9080	0.2721
6	32	50	0.9940	0.0293	0.9740	0.0975	0.9180	0.2463
7	64	10	0.9532	0.3329	0.9400	0.3791	0.8640	0.5758

Lanjutan Tabel 4.4

No	Batch Size	Epoch	Accuracy	Loss	Validasi Accuracy	Validasi Loss	Test Accuracy	Test Loss
8	64	30	0.9847	0.1059	0.9580	0.1611	0.9100	0.3131
9	64	50	0.9923	0.0583	0.9700	0.1198	0.9240	0.2813

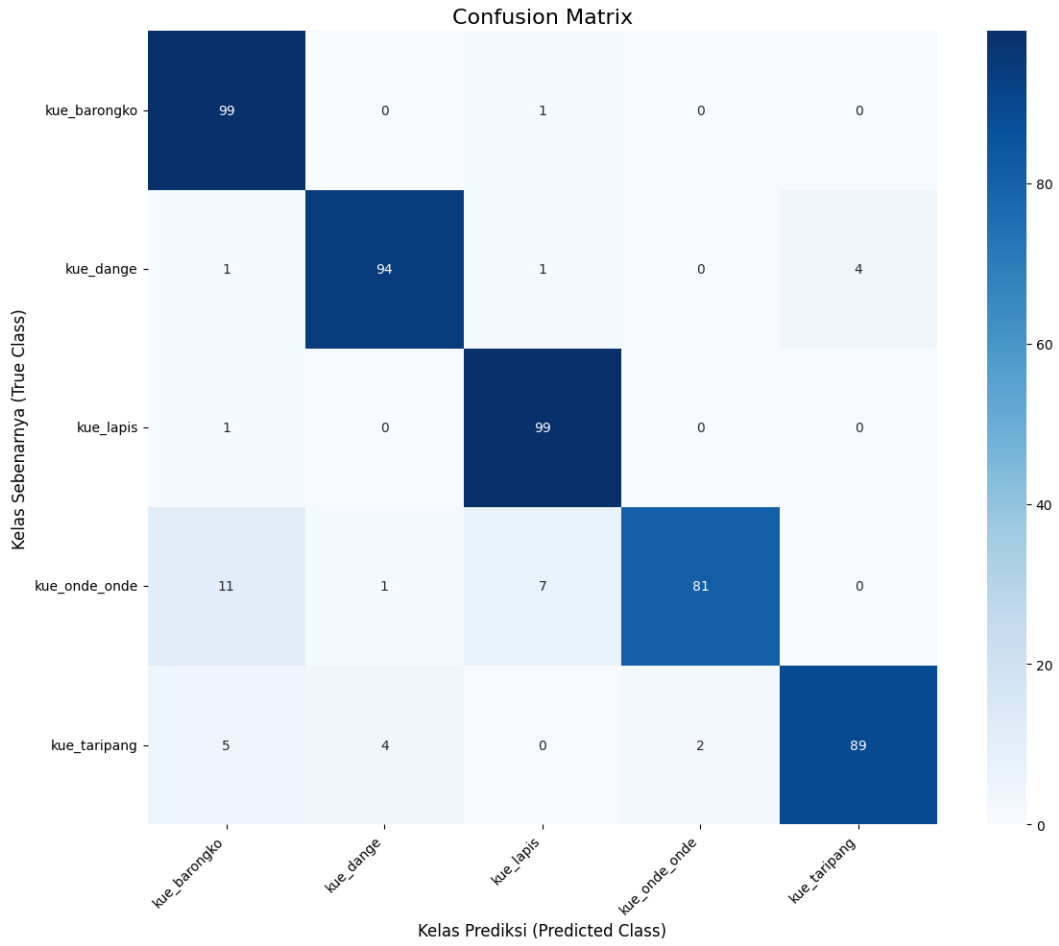
Berdasarkan tabel 4.4, pelatihan dengan batch size 64 dan epoch 50 menghasilkan akurasi terbaik pada model yang dikembangkan. Akurasi yang tercapai pada model ini adalah 0.9923 untuk akurasi pelatihan, 0.9700 untuk akurasi validasi, dan 0.9240 untuk akurasi testing. Grafik akurasi serta loss dari model ini bisa diamati di Gambar 4.21



Gambar 4.21 Hasil Grafik Akurasi dan Loss Model EfficientNetB0

Pengujian dalam penelitian ini dilakukan menggunakan metode *Confusion Matrix*, yang merupakan salah satu teknik evaluasi kinerja dari suatu model klasifikasi dalam *machine learning*. *Confusion matrix* menampilkan representasi terkait seberapa baik atau seberapa buruk model dalam melakukan klasifikasi pada dataset yang diberikan. Metode ini berguna untuk menghitung kllasifikasi, metode yang sangat berguna untuk mengutus recall, precision, akurasi, dan F1-score serta

memberikan informasi jumlah kelas prediksi yang diberikan. Evaluasi dilakukan berdasarkan model terbaik yang diperoleh dari proses pelatihan. Model yang diuji dalam penelitian ini adalah hasil pelatihan dengan menggunakan *batch size* 64 dan *epoch* 50, yang menghasilkan *training accuracy* sebesar 0.9821, *validation accuracy* sebesar 0.9560, dan *testing accuracy* sebesar 0.9000. Hasil pengujian penelitian berdasarkan model terbaik dapat dilihat pada Gambar 4.22 berikut:



Gambar 4.22 Evaluasi Confusion Matrix EfficientNetB0

Pada Gambar 4.22, dilakukan pengujian menggunakan *Confusion Matrix* untuk mengevaluasi kemampuan model dalam mengenali gambar berdasarkan

dataset uji yang telah disediakan dengan menerapkan metode CNN. Hasil pengujian menunjukkan nilai *accuracy, recall, precision*, dan *F1-score* sebagai berikut:

Parameter evaluasi ini dihitung berdasarkan jumlah *True Positive (TP), True Negative (TN), False Negative (FN), dan False Positive (FP)* untuk masing-masing kelas. Berikut hasil perhitungan dari Akurasi, *Precision, Recall* dan *F1-Score* berdasarkan setiap kelasnya:

a. Kelas Kue Barongko

   TP = 99, FP = 18, FN = 1, dan TN = 382

b. Kelas Kue Dange

   TP = 94, FP = 5, FN = 6, dan TN = 395

c. Kelas Kue Lapis

   TP = 99, FP = 9, FN = 1, dan TN = 391

d. Kelas Kue Onde-Onde

   TP = 81, FP = 2, FN = 19, dan TN = 398

e. Kelas Kue Taripang

   TP = 89, FP = 4, FN = 11, dan TN = 396

   Berikut dibawah ini rumus dan hasil dari Akurasi, Precision, Recall, dan F1 Score:

a. Akurasi

   Akurasi bisa diperoleh memakai formula

   Total data uji = 500

   Total Prediksi Benar = 99 + 94 + 99 + 81 + 89 = 462

   Akurasi = $\frac{450}{500} = 0.924\ atau\ 92.4\%$

b. Precision

1) Kue Barongko, Precision = $\frac{99}{99+18} = 0.8462$

2) Kue Dange, Precision = $\frac{94}{94+5} = 0.9495$

3) Kue Lapis, Precision = $\frac{99}{99+9} = 0.9167$

4) Kue Onde-Onde, Precision = $\frac{81}{81+2} = 0.9759$

5) Kue Taripang, Precision = $\frac{89}{89+4} = 0.9570$

c. Recall

1) Kue Barongko, Recall = $\frac{99}{99+1} = 0.99$

2) Kue Dange, Recall = $\frac{94}{94+6} = 0.94$

3) Kue Lapis, Recall = $\frac{99}{99+1} = 0.99$

4) Kue Onde-Onde, Recall = $\frac{81}{81+19} = 0.81$

5) Kue Taripang, Recall = $\frac{89}{89+11} = 0.89$

d. F1-Score

1) Kue Barongko, F1-Score $= 2\,x\ \frac{0.8462 \times 0.9900}{0.8462+ 0.9900} = 0.9122$

2) Kue Dange, F1-Score $= 2\,x\ \frac{0.9495 \times 0.9400}{0.9495+ 0.9400} = 0.9447$

3) Kue Lapis, F1-Score $= 2\,x\ \frac{0.9167 \times 0.9900}{0.9167+ 0.9900} = 0.9518$

4) Kue Onde-Onde, F1-Score $= 2\,x\ \frac{0.9759 \times 0.8100}{0.9759+ 0.8100} = 0.8856$

5) Kue Taripang, F1-Score $= 2\,x\ \frac{0.9570 \times 0.8900}{0.9570+ 0.8900} = 0.9223$

Berdasarkan evaluasi Confusion Matrix, model yang diuji menunjukkan performa yang sangat impresif dengan akurasi keseluruhan mencapai 92.4%. Akurasi yang sangat tinggi ini menegaskan bahwa metode CNN yang digunakan sangat efektif dalam tugas klasifikasi citra kue tradisional.

Analisis lebih lanjut menunjukkan kinerja yang luar biasa pada beberapa kelas. Kue Lapis menjadi kelas dengan performa terbaik, mencapai F1-Score tertinggi (0.9518) dan Recall nyaris sempurna (0.99). Kue Dange juga menunjukkan kinerja yang sangat seimbang dan solid dengan F1-Score 0.9447. Tantangan utama model terlihat pada Kue Onde-Onde, yang memiliki Recall terendah (0.81), menandakan model masih sering melewatkan kelas ini meskipun presisinya sangat tinggi (0.9759). Sebaliknya, Kue Barongko memiliki Precision terendah (0.8462), menunjukkan kecenderungan model untuk salah mengklasifikasikan kue lain sebagai Barongko. Secara keseluruhan, hasil ini memvalidasi kemampuan model yang sangat kuat, dengan wawasan spesifik pada setiap kelas untuk potensi perbaikan di masa depan.

Tiga arsitektur CNN, yaitu MobileNetV2, MobileNetV3Small, dan EfficientNet-B0, dievaluasi untuk tugas klasifikasi citra kue tradisional Bugis. Untuk menemukan konfigurasi optimal, setiap arsitektur dilatih dengan memvariasikan hyperparameter batch size (16, 32, 64) dan jumlah epoch (10, 30, 50). Kinerja dievaluasi berdasarkan akurasi pada data pelatihan, validasi, dan pengujian. Tabel 4.5 menyajikan perbandingan kinerja optimal dari setiap arsitektur setelah melalui proses hyperparameter tuning.

Tabel 4.5 Perbandingan Kinerja Optimal dari Setiap Arsitektur

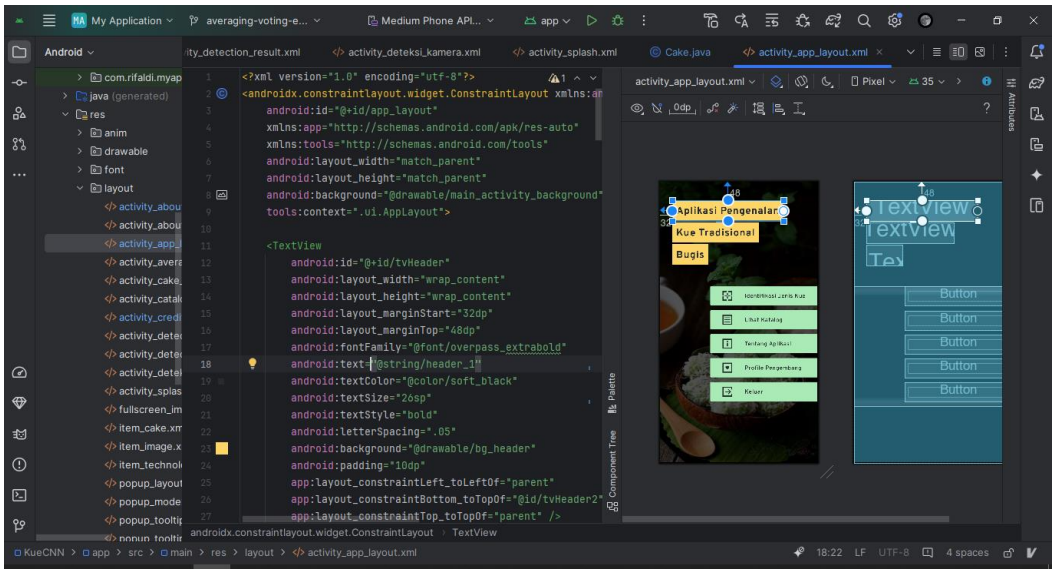
Arsitektur	Batch Size	Epoch	Akurasi Pelatihan	Akurasi Validasi	Akurasi Pengujian
MobileNetV2	32	50	0.9957	0.9620	0.9060
MobileNetV3Small	16	50	0.9949	0.9520	0.8740
EfficientNet-B0	64	50	0.9923	0.9700	0.9240

Hasil pada Tabel 4.5 menunjukkan bahwa model EfficientNet-B0 dengan konfigurasi batch size 64 dan 50 epoch memberikan performa paling unggul dan stabil. Model ini mencapai akurasi pengujian tertinggi sebesar 92.4% (0.9240). Berdasarkan hasil tersebut, EfficientNet-B0 dipilih untuk sebagai model yang akan diimplementasikan pada perangkat Android.

G. Perancangan Sistem

1. Proses Pembuatan *User Interface*

Berikut tampilan dan penjelasan mengenai proses pembuatan *user interface* yang ada ada aplikasi:



Gambar 4.23 Pembuatan *User Interface* Aplikasi

Antarmuka pengguna (UI) aplikasi dirancang menggunakan *layout* berbasis XML di Android Studio. Gambar 4.23 menampilkan proses desain untuk halaman menu utama (activity_app_layout.xml).

Pada proses ini, <ConstraintLayout> digunakan sebagai *layout* utama untuk menata posisi komponen secara responsif. Komponen-komponen visual seperti <TextView> untuk menampilkan teks dan <Button> untuk interaksi pengguna didefinisikan di dalam file XML.

Meskipun gambar ini hanya menampilkan perancangan untuk satu halaman, metodologi yang sama—meliputi pendefinisian dalam XML, penataan dengan ConstraintLayout, dan pemanfaatan sumber daya terpusat—diterapkan secara konsisten pada seluruh layout lainnya untuk membangun antarmuka pengguna aplikasi yang utuh dan kohesif.

2. Proses Inference Model

```
import com.rifaldi.myapplication.R;
import com.rifaldi.myapplication.ml.MobileNetV2;
import com.rifaldi.myapplication.ml.BestEfficientNetB0;
import com.rifaldi.myapplication.ml.BestMobileNetV3Small;
import org.tensorflow.lite.DataType;
import org.tensorflow.lite.support.common.ops.DequantizeOp;
import org.tensorflow.lite.support.common.ops.NormalizeOp;
import org.tensorflow.lite.support.image.ImageProcessor;
import org.tensorflow.lite.support.image.TensorImage;
import org.tensorflow.lite.support.image.ops.ResizeOp;
import org.tensorflow.lite.support.tensorbuffer.TensorBuffer;
```

Bagian ini menampilkan deklarasi import yang menjadi fondasi dari proses inferensi. Pustaka yang digunakan mencakup komponen inti TensorFlow Lite (org.tensorflow.lite) untuk pemrosesan tensor, serta kelas-kelas model yang digenerasi secara otomatis oleh fitur ML Model Binding (com.rifaldi.myapplication.ml.*) yang mempermudah interaksi dengan file .tflite.

```
private float[] getModelProbabilities(Bitmap bitmap, String
modelType) throws IOException {
    try {
        TensorImage tensorImage = preprocessImage(bitmap,
modelType);

        switch (modelType) {
            case "MobileNetV3": {
                BestMobileNetV3Small cake_model =
BestMobileNetV3Small.newInstance(getApplicationContext());
                BestMobileNetV3Small.Outputs outputs =
cake_model.process(tensorImage.getTensorBuffer());
                TensorBuffer probabilityBuffer =
outputs.getOutputFeature0AsTensorBuffer();
                float[] probabilities =
probabilityBuffer.getFloatArray();
                cake_model.close();
                return probabilities;
            }
            case "MobileNetV2": {
                MobileNetV2 cake_model =
MobileNetV2.newInstance(getApplicationContext());
                MobileNetV2.Outputs outputs =
cake_model.process(tensorImage.getTensorBuffer());
                TensorBuffer probabilityBuffer =
outputs.getOutputFeature0AsTensorBuffer();
                float[] probabilities =
probabilityBuffer.getFloatArray();
                cake_model.close();
                return probabilities;

            }
            case "EfficientNetB0": {
                BestEfficientNetB0 cake_model =
BestEfficientNetB0.newInstance(getApplicationContext());
                BestEfficientNetB0.Outputs outputs =
cake_model.process(tensorImage.getTensorBuffer());
                TensorBuffer probabilityBuffer =
outputs.getOutputFeature0AsTensorBuffer();
                float[] probabilities =
probabilityBuffer.getFloatArray();
                cake_model.close();
                return probabilities;
            }
            default:
                Log.e("getModelProbabilites() ", "Model type
unknown");
                throw new IOException("Model type tidak
dikenal: " + modelType);
        }
```

```
    } catch (IOException e) {
        Log.e("getModelProbabilites() ", "Failed to load
model", e);
        throw new IOException("Gagal mendapatkan probabilitas
model: " + e.getMessage());
    }
}
```

Potongan kode ini menampilkan fungsi inti getModelProbabilities. Prosesnya mencakup instansiasi model (contoh: BestEfficientNetB0) menggunakan fitur *ML Model Binding*, eksekusi inferensi melalui metode .process(), pengambilan hasil prediksi dalam bentuk *array* probabilitas, dan pelepasan sumber daya dengan memanggil .close() untuk manajemen memori.

```
private void classifyImage(Bitmap bitmap){
    float[] firstModelProbabilities, secondModelProbabilities;
    try {
        firstModelProbabilities =
getModelProbabilities(bitmap, "EfficientNetB0");
        secondModelProbabilities =
getModelProbabilities(bitmap, "EfficientNetB0");

        String firstModelResult =
getMaxProbabilityLabel(firstModelProbabilities);
        String secondModelResult =
getMaxProbabilityLabel(secondModelProbabilities);
        String averagingResult =
averagingScore(firstModelProbabilities,
secondModelProbabilities);
        String detailedAveragingOutput =
detailedAveragingOutput(firstModelProbabilities,
secondModelProbabilities);

        prepareResultAndIntent(bitmap, firstModelResult,
secondModelResult, averagingResult, detailedAveragingOutput);

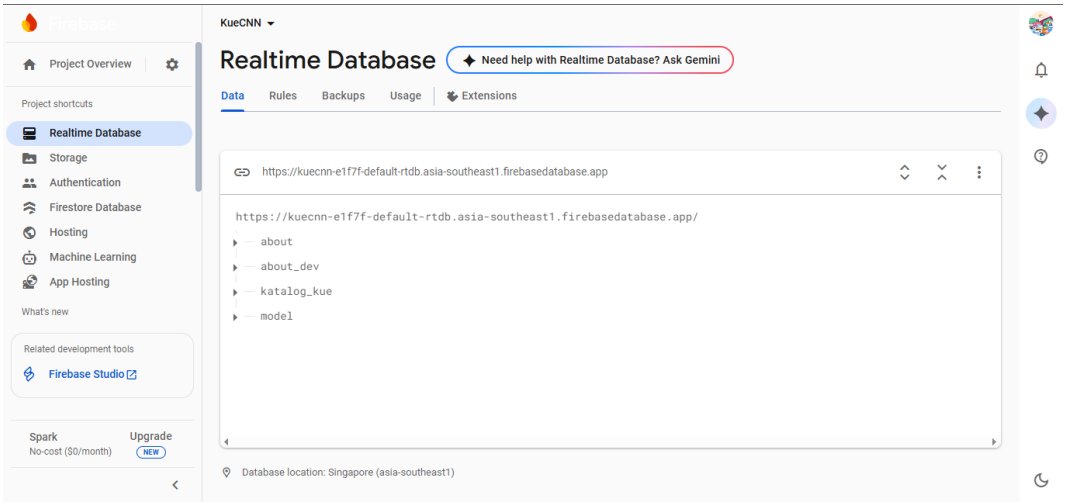
    } catch (IOException e) {
        throw new RuntimeException(e);
    }
}
```

Kode ini menunjukkan fungsi classifyImage yang mengorkestrasi seluruh proses. Setelah memanggil fungsi inferensi, hasilnya (nama kue dan skor) beserta gambar input akan dikemas ke dalam sebuah Intent oleh fungsi

prepareResultAndIntent. Intent ini kemudian digunakan untuk menampilkan informasi secara visual pada halaman hasil deteksi.

3. Proses Pembuatan Database

Pembuatan database menggunakan *Firebase* dengan format JSON sebagai berikut:



Gambar 4.24 Tampilan Desain Database Firebase

Pembuatan basis data untuk aplikasi ini memanfaatkan layanan *cloud* dari Google, yaitu Firebase Realtime Database. Platform ini dipilih karena kemudahannya dalam implementasi, skalabilitas, dan kemampuannya untuk menyinkronkan data secara *real-time.*

Firebase Realtime Database menggunakan struktur data NoSQL yang disimpan dalam format JSON, menyerupai sebuah pohon (*tree*). Struktur ini terdiri dari beberapa *node* (simpul) utama:

a. katalog_kue: Menyimpan seluruh data detail mengenai setiap kue tradisional, termasuk nama, deskripsi, resep, URL gambar, dan referensi.

b. about_dev: Berisi informasi statis mengenai pengembang aplikasi.

c. about: Menyimpan deskripsi umum mengenai aplikasi.

d. model: Menyimpan metadata terkait model CNN yang digunakan, seperti nama dan akurasi.

H. Implementasi

Implementasi sistem ialah upaya menerapkan teknologi yang telah didesain serta sedia digunakan oleh pengguna. Sistem ini dikembangkan menggunakan Python sebagai bahasa pemrograman untuk pembuatan model, sementara Dart digunakan dalam pengembangan aplikasi Android.

1. *Hardware* (Perangkat Keras)

Piranti keras yang dimanfaatkan guna mengembangkan aplikasi ini tercamtun pada tabel 4.6:

Tabel 4.6 Perangkat Keras (*Hardware*)

No	Spesifikasi	
1	Merk	Asus
2	Processor	Intel Core i3 6006U
3	Memory	8GB (Dual Slot)
4	Storage	SSD 512GB

2. *Software* (Perangkat Lunak)

Piranti lunak yang dimanfaatkan guna membangun aplikasi termaktub dalam tabel 4.7

Tabel 4.7 Perangkat Lunak (*Software*)

No	Spesifikasi	
1	Sistem Operasi (OS)	Windows 10
2	Aplikasi Membuat Model CNN	Google Chrome (Google Colab)
3	Aplikasi Membuat Aplikasi Android	Android Studio

3. Perangkat *Mobile* (Android)

Piranti genggam yang diperuntukkan bagi pengujian aplikasi tertera pada tabel 4.8.

Tabel 4.8 Perangkat *Mobile*

No	Spesifikasi	
1	Merk	Realme 3 Pro
2	Sistem Operasi	Android 10
3	Processor	Qualcomm Snapdragon 710
4	RAM/ROM	4GB/64GB
5	Resolusi Layar	1080 x 2340 piksel
6	Baterai	4045mAh

I. Pengujian Sistem

Dalam pengujian aplikasi ini, terdapat 2 metode utama, yaitu *Black Box* dan *White Box* serta pengujian tambahan yaitu pengujian fitur deteksi. Uji coba sistem ini tertuju untuk mengetahui proses dan cara kerja aplikasi.

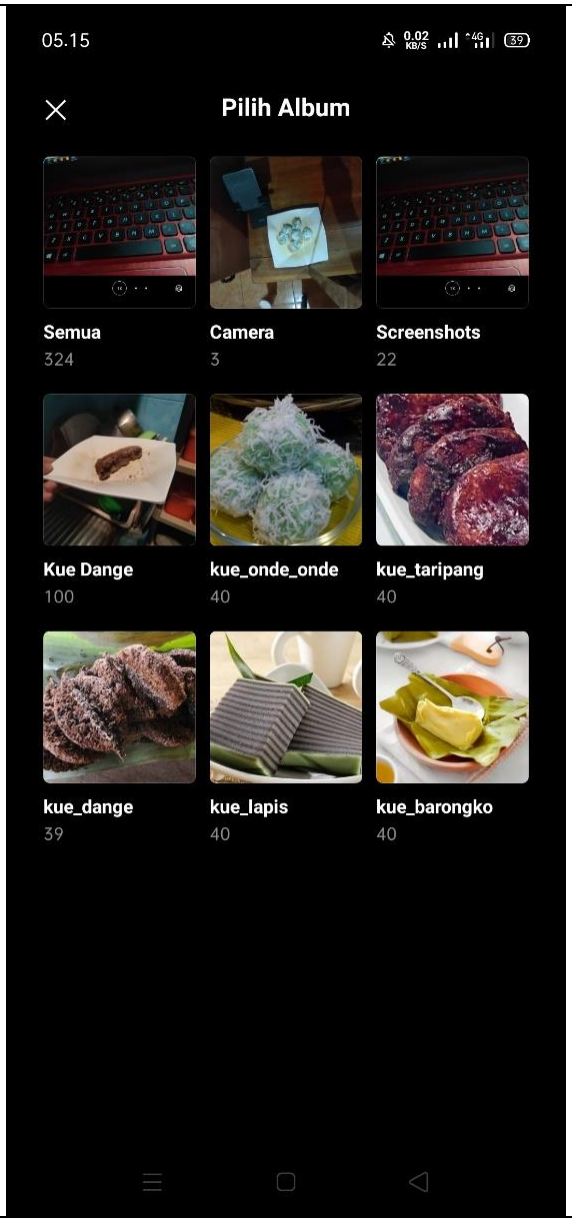
1. Pengujian *Black Box*

Berikut adalah pengujian *Black Box* yang diterapkan pada aplikasi:

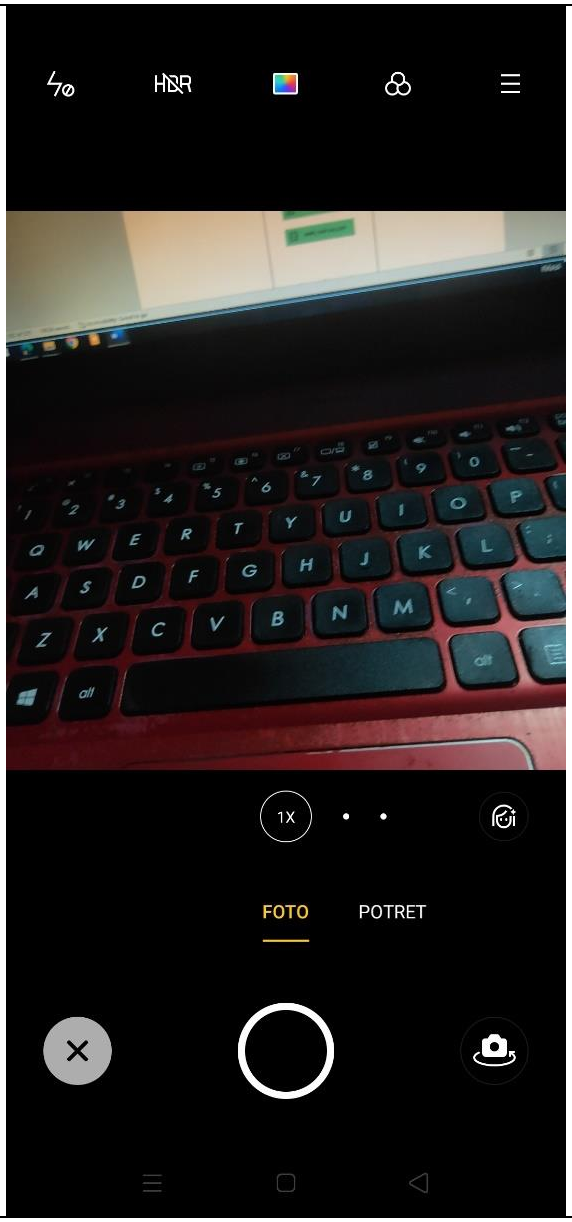
Tabel 4.9 Halaman Depan

No	Pengujian	Test Factor	Hasil	Keterangan
1	Pengujian Halaman depan	Menunjukkan halaman depan aplikasi	Berhasil	Berhasil menampilkan halaman depan
Screenshot				
				

Tabel 4.10 Tombol Ambil *Gallery*

No	Pengujian	Test Factor	Hasil	Keterangan
1	Pengujian Halaman ambil galeri	Menginput gambar melalui galeri handphone	Berhasil memiliih gambar	Berhasil memasukkan gambar yang dari galeri
Screenshot				
				

Tabel 4.11 Tombol Ambil Kamera

No	Pengujian	Test Factor	Hasil	Keterangan
1	Pengujian Halaman ambil gambar	Mengambil gambar melalui handphone	Berhasil mengambil gambar	Berhasil mengambil gambar melalui handphone
Screenshot				
				

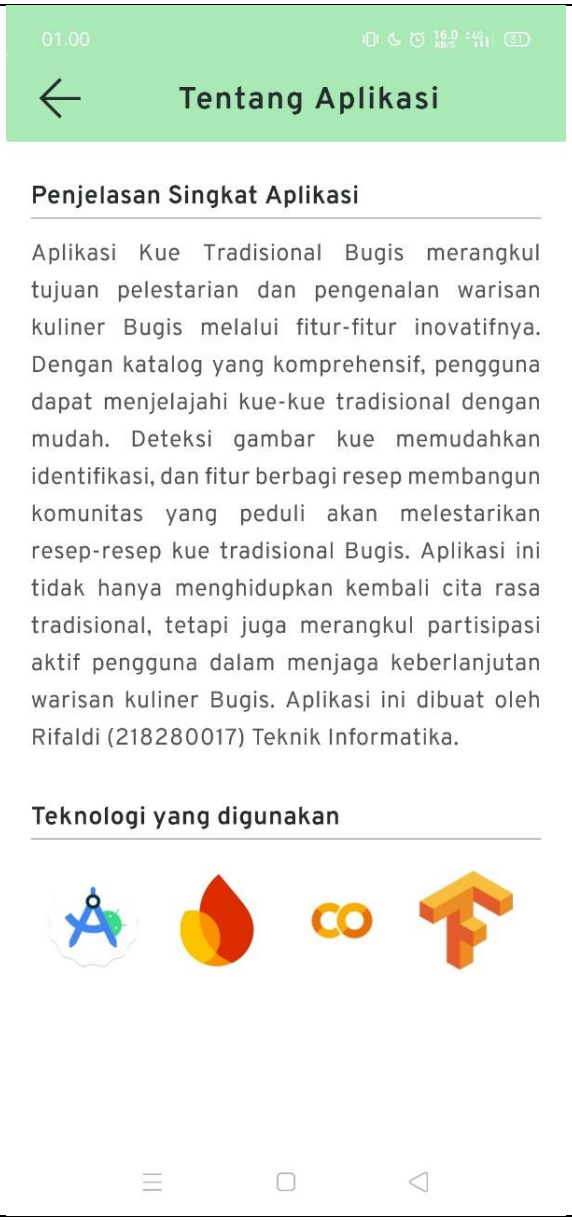
Tabel 4.12 Halaman Katalog Kue Tradisional Bugis

No	Pengujian	Test Factor	Hasil	Keterangan
1	Pengujian Halaman katalog kue tradisional Bugis	Menampilkan halaman katalog kue	Berhasil menampilkan	Berhasil menampilkan halaman katalog kue
Screenshot				



Tabel 4.13 Halaman Tentang Aplikasi

No	Pengujian	Test Factor	Hasil	Keterangan
1	Pengujian Halaman tentang aplikasi	Menampilkan halaman tentang aplikasi	Berhasil menampilkan	Berhasil menampilkan halaman aplikasi
Screenshot				



Tabel 4.14 Halaman Hasil Deteksi Kue

No	Pengujian	Test Factor	Hasil	Keterangan
1	Pengujian deteksi kue	Menampilkan gambar kue yang dimasukkan	Berhasil mendeteksi	Berhasil mendeteksi kue yang di masukkan
Screenshot				
				

Tabel 4.15 Halaman Deskripsi Kue

No	Pengujian	Test Factor	Hasil	Keterangan
1	Pengujian halaman deskripsi kue	Menampilkan deskripsi kue yang dipilih	Berhasil menampilkan	Berhasil menampilkan deskripsi kue
Screenshot				

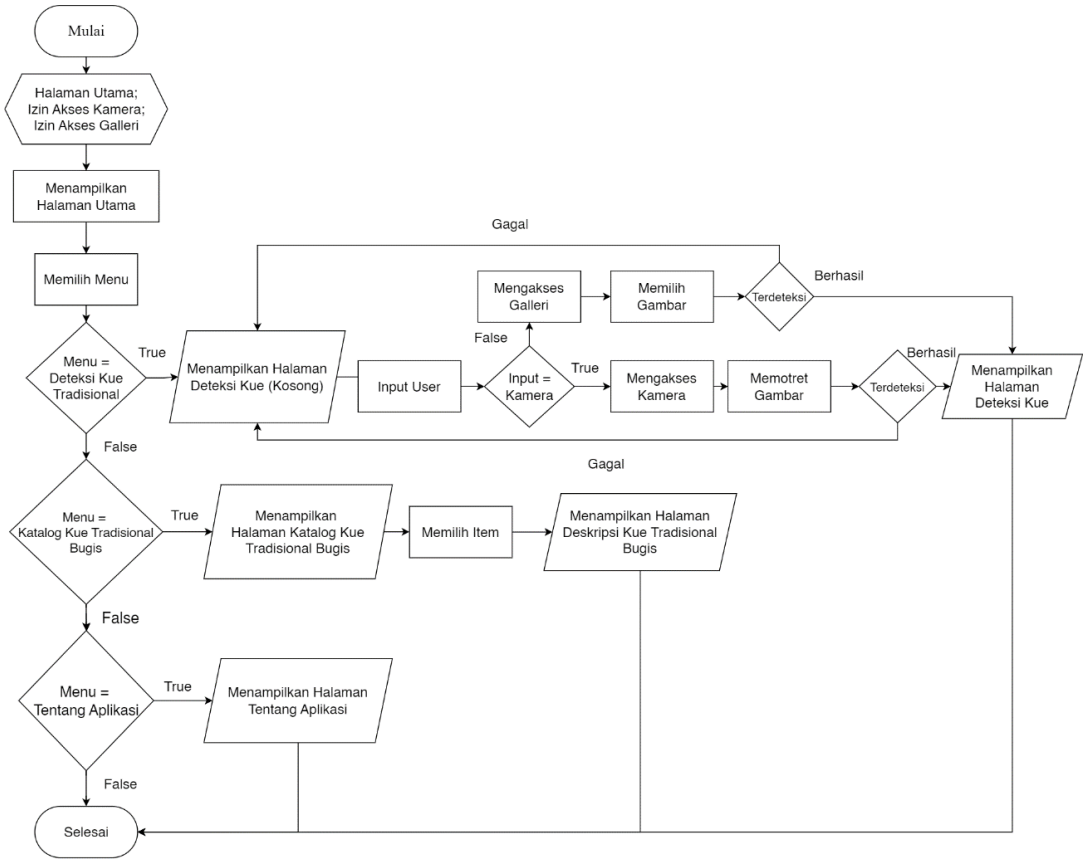


Metode pengujian langsung berdasarkan teknik *black-box* sangat efektif dalam menguji fungsionalitas aplikasi secara menyeluruh. Dengan menguji tombol dan hasil aplikasi, proses ini memastikan bahwa setiap aspek dari aplikasi berjalan sesuai dengan yang diharapkan. Hasil pengujian ini membantu memastikan kualitas dan keandalan sistem sebelum dirilis kepada pengguna.

2. Pengujian *White Box*

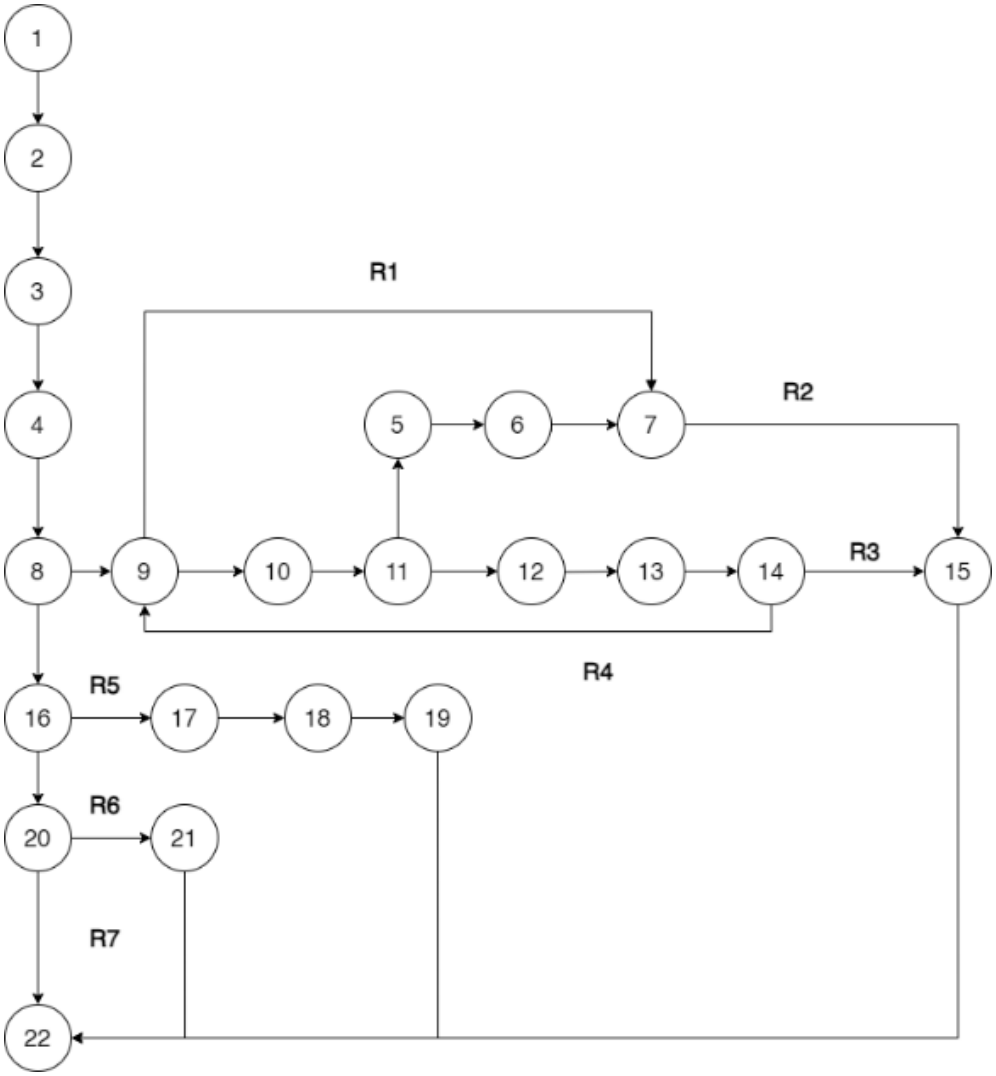
Berikut adalah pengujian *White Box* yang diterapkan pada aplikasi:

a. *Flowchart* Aplikasi



Gambar 4.25 *Flowchart* Aplikasi

b. *Flowgraph* Aplikasi



Gambar 4.26 *Flowgraph* Aplikasi

Dari *flowgraph* yang disajikan, kalkulasi bisa dilakukan seperti berikut:

1) Menentukan *Cyclomatic Complexity* V(G) berdasarkan jumlah *Edge* dan *Node* menggunakan rumus yang sesuai:

V(G) = E – N + 2

N(node) = 22

E (edge) = 27

P(Predikat node) = 6

Penyelesaian

V(G) = E – N +2

= 27 - 22 + 2

= 7

Predikat (P) = P + 1

= 6 + 1

= 7

2) Berlandaskan analisis kalkulasi *Cyclomatic Complexcity flowgraph* di atas, flowgraph tersebut memiliki jumlah *Region* (R) sebesar 7.

3) *Independent path flowgraph* di atas adalah:

Path 1: 1, 2, 3, 4, 8, 9, 10, 11, 5, 6, 7, 9, 10, 11, 5, 6, 7, 15, 22.

Path 2: 1, 2, 3, 4, 8, 9, 10, 11, 5, 6, 7, 15, 22

Path 3: 1, 2, 3, 4, 8, 9, 10, 11, 12, 13, 14, 15, 22

Path 4: 1, 2, 3, 4, 8, 9, 10, 11, 12, 13, 14, 9, 10, 11, 12, 13, 14, 15, 22

Path 5: 1, 2, 3, 4, 8, 16, 17, 18, 19, 22

Path 6: 1, 2, 3, 4, 8, 16, 20, 21, 22

Path 7: 1, 2, 3, 4, 8, 16, 20, 22

c. Grafik matriks aplikasi

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	E-1
1		1																					1-1=0
2			1																				1-1=0
3				1																			1-1=0
4								1															1-1=0
5						1																	1-1=0
6							1																1-1=0
7									1						1								2-1=1
8									1							1							2-1=1
9										1													1-1=0
10											1												1-1=0
11					1							1											2-1=1
12													1										1-1=0
13														1									1-1=0
14									1						1								2-1=1
15																						1	1-1=0
16																	1			1			2-1=1
17																		1					1-1=0
18																			1				1-1=0
19																						1	1-1=0
20																					1	1	2-1=1
21																					1		1-1=0
22																							0
SUM (E+1)																							6+1=7

3. Pengujian Fitur Deteksi

Berikut ini proses pengujian fitur utama dari aplikasi yaitu Deteksi Kue Tradisional Bugis dengan jarak ukur, kemiringan, jumlah kue, dan kondisi kue.

Semua gambar diambil menggunakan kamera Realme 3 Pro untuk memastikan konsistensi sensor, lensa, dan pemrosesan gambar internal. Berikut tabel pengujiannya fitur tersebut:

Tabel 4.16 Tabel Pengujian Fitur Deteksi

No	Nama Kue	Jarak (cm)	Kemiringan (derajat)	Jumlah	Kondisi	Hasil
1	Kue Barongko	15 cm	0°	1 kue	Tertutup	 Berhasil ✓
2	Kue Barongko	20 cm	0°	1 kue	Tertutup	 Berhasil ✓

Lanjutan Tabel 4.16

No	Nama Kue	Jarak (cm)	Kemiringan (derajat)	Jumlah	Kondisi	Hasil
3	Kue Barongko	30 cm	0°	1 kue	Tertutup	 Berhasil ✓
4	Kue Barongko	15 cm	45°	1 kue	Tertutup	 Berhasil ✓
5	Kue Barongko	20 cm	45°	1 kue	Tertutup	 Berhasil ✓

Lanjutan Tabel 4.16

No	Nama Kue	Jarak (cm)	Kemiringan (derajat)	Jumlah	Kondisi	Hasil
6	Kue Barongko	30 cm	45°	1 kue	Tertutup	 Berhasil ✓
7	Kue Barongko	15 cm	90°	1 kue	Tertutup	 Berhasil ✓
8	Kue Barongko	20 cm	90°	1 kue	Tertutup	 Berhasil ✓

Lanjutan Tabel 4.16

No	Nama Kue	Jarak (cm)	Kemiringan (derajat)	Jumlah	Kondisi	Hasil
9	Kue Barongko	30 cm	90°	1 kue	Tertutup	 Berhasil ✓
10	Kue Barongko	15 cm	0°	2 kue	Tertutup	 Berhasil ✓
11	Kue Barongko	20 cm	0°	2 kue	Tertutup	 Berhasil ✓

Lanjutan Tabel 4.16

No	Nama Kue	Jarak (cm)	Kemiringan (derajat)	Jumlah	Kondisi	Hasil
12	Kue Barongko	30 cm	0°	2 kue	Tertutup	 Berhasil ✓
13	Kue Barongko	15 cm	45°	2 kue	Tertutup	 Berhasil ✓
14	Kue Barongko	20 cm	45°	2 kue	Tertutup	 Berhasil ✓

Lanjutan Tabel 4.16

No	Nama Kue	Jarak (cm)	Kemiringan (derajat)	Jumlah	Kondisi	Hasil
15	Kue Barongko	30 cm	45°	2 kue	Tertutup	 Berhasil ✓
16	Kue Barongko	15 cm	90°	2 kue	Tertutup	 Berhasil ✓
17	Kue Barongko	20 cm	90°	2 kue	Tertutup	 Berhasil ✓

Lanjutan Tabel 4.16

No	Nama Kue	Jarak (cm)	Kemiringan (Derajat)	Jumlah	Kondisi	Hasil
18	Kue Barongko	30 cm	90°	2 kue	Tertutup	 Berhasil ✓
19	Kue Barongko	15 cm	0°	1 kue	Terbuka	 Berhasil ✓
20	Kue Barongko	20 cm	0°	1 kue	Terbuka	 Berhasil ✓

Lanjutan Tabel 4.16

No	Nama Kue	Jarak (cm)	Kemiringan (Derajat)	Jumlah	Kondisi	Hasil
21	Kue Barongko	30 cm	0°	1 kue	Terbuka	 Berhasil ✓
22	Kue Barongko	15 cm	45°	1 kue	Terbuka	 Berhasil ✓
23	Kue Barongko	20 cm	45°	1 kue	Terbuka	 Berhasil ✓

Lanjutan Tabel 4.16

No	Nama Kue	Jarak (cm)	Kemiringan (Derajat)	Jumlah	Kondisi	Hasil
24	Kue Barongko	30 cm	45°	1 kue	Terbuka	 Berhasil ✓
25	Kue Barongko	15 cm	90°	1 kue	Terbuka	 Berhasil ✓
26	Kue Barongko	20 cm	90°	1 kue	Terbuka	 Berhasil ✓

Lanjutan Tabel 4.16

No	Nama Kue	Jarak (cm)	Kemiringan (Derajat)	Jumlah	Kondisi	Hasil
27	Kue Barongko	30 cm	90°	1 kue	Terbuka	 Berhasil ✓
28	Kue Barongko	15 cm	0°	2 kue	Terbuka	 Berhasil ✓
29	Kue Barongko	20 cm	0°	2 kue	Terbuka	 Berhasil ✓

Lanjutan Tabel 4.16

No	Nama Kue	Jarak (cm)	Kemiringan (Derajat)	Jumlah	Kondisi	Hasil
30	Kue Barongko	30 cm	0°	2 kue	Terbuka	 Berhasil ✓
31	Kue Barongko	15 cm	45°	2 kue	Terbuka	 Berhasil ✓
32	Kue Barongko	20 cm	45°	2 kue	Terbuka	 Berhasil ✓

Lanjutan Tabel 4.16

No	Nama Kue	Jarak (cm)	Kemiringan (Derajat)	Jumlah	Kondisi	Hasil
33	Kue Barongko	30 cm	45°	2 kue	Terbuka	 Berhasil ✓
34	Kue Barongko	15 cm	90°	2 kue	Terbuka	 Berhasil ✓
35	Kue Barongko	20 cm	90°	2 kue	Terbuka	 Berhasil ✓

Lanjutan Tabel 4.16

No	Nama Kue	Jarak (cm)	Kemiringan (Derajat)	Jumlah	Kondisi	Hasil
36	Kue Barongko	30 cm	90°	2 kue	Terbuka	 Berhasil ✓
37	Kue Dange	15 cm	0°	1 kue	Utuh	 Berhasil ✓
38	Kue Dange	20 cm	0°	1 kue	Utuh	 Berhasil ✓

Lanjutan Tabel 4.16

No	Nama Kue	Jarak (cm)	Kemiringan (Derajat)	Jumlah	Kondisi	Hasil
39	Kue Dange	30 cm	0°	1 kue	Utuh	 Berhasil ✓
40	Kue Dange	15 cm	45°	1 kue	Utuh	 Berhasil ✓
41	Kue Dange	20 cm	45°	1 kue	Utuh	 Berhasil ✓

Lanjutan Tabel 4.16

No	Nama Kue	Jarak (cm)	Kemiringan (Derajat)	Jumlah	Kondisi	Hasil
42	Kue Dange	30 cm	45°	1 kue	Utuh	 Berhasil ✓
43	Kue Dange	15 cm	90°	1 kue	Utuh	 Berhasil ✓
44	Kue Dange	20 cm	90°	1 kue	Utuh	 Berhasil ✓

Lanjutan Tabel 4.16

No	Nama Kue	Jarak (cm)	Kemiringan (Derajat)	Jumlah	Kondisi	Hasil
45	Kue Dange	30 cm	90°	1 kue	Utuh	 Berhasil ✓
46	Kue Dange	15 cm	0°	2 kue	Utuh	 Berhasil ✓
47	Kue Dange	20 cm	0°	2 kue	Utuh	 Berhasil ✓

Lanjutan Tabel 4.16

No	Nama Kue	Jarak (cm)	Kemiringan (Derajat)	Jumlah	Kondisi	Hasil
48	Kue Dange	30 cm	0°	2 kue	Utuh	 Berhasil ✓
49	Kue Dange	15 cm	45°	2 kue	Utuh	 Berhasil ✓
50	Kue Dange	20 cm	45°	2 kue	Utuh	 Berhasil ✓

Lanjutan Tabel 4.16

No	Nama Kue	Jarak (cm)	Kemiringan (Derajat)	Jumlah	Kondisi	Hasil
51	Kue Dange	30 cm	45°	2 kue	Utuh	 Berhasil ✓
52	Kue Dange	15 cm	90°	2 kue	Utuh	 Berhasil ✓
53	Kue Dange	20 cm	90°	2 kue	Utuh	 Berhasil ✓

Lanjutan Tabel 4.16

No	Nama Kue	Jarak (cm)	Kemiringan (Derajat)	Jumlah	Kondisi	Hasil
54	Kue Dange	30 cm	90°	2 kue	Utuh	 Berhasil ✓
55	Kue Dange	15 cm	0°	1 kue	Terpotong	 Berhasil ✓
56	Kue Dange	20 cm	0°	1 kue	Terpotong	 Gagal ×

Lanjutan Tabel 4.16

No	Nama Kue	Jarak (cm)	Kemiringan (Derajat)	Jumlah	Kondisi	Hasil
57	Kue Dange	30 cm	0°	1 kue	Terpotong	 Gagal ×
58	Kue Dange	15 cm	45°	1 kue	Terpotong	 Berhasil ✓
59	Kue Dange	20 cm	45°	1 kue	Terpotong	 Berhasil ✓

Lanjutan Tabel 4.16

No	Nama Kue	Jarak (cm)	Kemiringan (Derajat)	Jumlah	Kondisi	Hasil
60	Kue Dange	30 cm	45°	1 kue	Terpotong	 Gagal ×
61	Kue Dange	15 cm	90°	1 kue	Terpotong	 Berhasil ✓
62	Kue Dange	20 cm	90°	1 kue	Terpotong	 Berhasil ✓

Lanjutan Tabel 4.16

No	Nama Kue	Jarak (cm)	Kemiringan (Derajat)	Jumlah	Kondisi	Hasil
63	Kue Dange	30 cm	90°	1 kue	Terpotong	 Berhasil ✓
64	Kue Dange	15 cm	0°	2 kue	Terpotong	 Berhasil ✓
65	Kue Dange	20 cm	0°	2 kue	Terpotong	 Berhasil ✓

Lanjutan Tabel 4.16

No	Nama Kue	Jarak (cm)	Kemiringan (Derajat)	Jumlah	Kondisi	Hasil
66	Kue Dange	30 cm	0°	2 kue	Terpotong	 Berhasil ✓
67	Kue Dange	15 cm	45°	2 kue	Terpotong	 Berhasil ✓
68	Kue Dange	20 cm	45°	2 kue	Terpotong	 Berhasil ✓

Lanjutan Tabel 4.16

No	Nama Kue	Jarak (cm)	Kemiringan (Derajat)	Jumlah	Kondisi	Hasil
69	Kue Dange	30 cm	45°	2 kue	Terpotong	 Berhasil ✓
70	Kue Dange	15 cm	90°	2 kue	Terpotong	 Berhasil ✓
71	Kue Dange	20 cm	90°	2 kue	Terpotong	 Gagal ×

Lanjutan Tabel 4.16

No	Nama Kue	Jarak (cm)	Kemiringan (Derajat)	Jumlah	Kondisi	Hasil
72	Kue Dange	30 cm	90°	2 kue	Terpotong	 Gagal ×
73	Kue Lapis	15 cm	0°	1 kue	Utuh	 Berhasil ✓
74	Kue Lapis	20 cm	0°	1 kue	Utuh	 Berhasil ✓

Lanjutan Tabel 4.16

No	Nama Kue	Jarak (cm)	Kemiringan (Derajat)	Jumlah	Kondisi	Hasil
75	Kue Lapis	30 cm	0°	1 kue	Utuh	 Berhasil ✓
76	Kue Lapis	15 cm	45°	1 kue	Utuh	 Berhasil ✓
77	Kue Lapis	20 cm	45°	1 kue	Utuh	 Berhasil ✓

Lanjutan Tabel 4.16

No	Nama Kue	Jarak (cm)	Kemiringan (Derajat)	Jumlah	Kondisi	Hasil
78	Kue Lapis	30 cm	45°	1 kue	Utuh	 Berhasil ✓
79	Kue Lapis	15 cm	90°	1 kue	Utuh	 Berhasil ✓
80	Kue Lapis	20 cm	90°	1 kue	Utuh	 Berhasil ✓

Lanjutan Tabel 4.16

No	Nama Kue	Jarak (cm)	Kemiringan (Derajat)	Jumlah	Kondisi	Hasil
81	Kue Lapis	30 cm	90°	1 kue	Utuh	 Berhasil ✓
82	Kue Lapis	15 cm	0°	2 kue	Utuh	 Berhasil ✓
83	Kue Lapis	20 cm	0°	2 kue	Utuh	 Berhasil ✓

Lanjutan Tabel 4.16

No	Nama Kue	Jarak (cm)	Kemiringan (Derajat)	Jumlah	Kondisi	Hasil
84	Kue Lapis	30 cm	0°	2 kue	Utuh	 Gagal ×
85	Kue Lapis	15 cm	45°	2 kue	Utuh	 Berhasil ✓
86	Kue Lapis	20 cm	45°	2 kue	Utuh	 Berhasil ✓

Lanjutan Tabel 4.16

No	Nama Kue	Jarak (cm)	Kemiringan (Derajat)	Jumlah	Kondisi	Hasil
87	Kue Lapis	30 cm	45°	2 kue	Utuh	 Berhasil ✓
88	Kue Lapis	15 cm	90°	2 kue	Utuh	 Berhasil ✓
89	Kue Lapis	20 cm	90°	2 kue	Utuh	 Berhasil ✓

Lanjutan Tabel 4.16

No	Nama Kue	Jarak (cm)	Kemiringan (Derajat)	Jumlah	Kondisi	Hasil
90	Kue Lapis	30 cm	90°	2 kue	Utuh	 Berhasil ✓
91	Kue Lapis	15 cm	0°	1 kue	Terpotong	 Berhasil ✓
92	Kue Lapis	20 cm	0°	1 kue	Terpotong	 Berhasil ✓

Lanjutan Tabel 4.16

No	Nama Kue	Jarak (cm)	Kemiringan (Derajat)	Jumlah	Kondisi	Hasil
93	Kue Lapis	30 cm	0°	1 kue	Terpotong	 Gagal ×
94	Kue Lapis	15 cm	45°	1 kue	Terpotong	 Berhasil ✓
95	Kue Lapis	20 cm	45°	1 kue	Terpotong	 Berhasil ✓

Lanjutan Tabel 4.16

No	Nama Kue	Jarak (cm)	Kemiringan (Derajat)	Jumlah	Kondisi	Hasil
96	Kue Lapis	30 cm	45°	1 kue	Terpotong	 Berhasil ✓
97	Kue Lapis	15 cm	90°	1 kue	Terpotong	 Berhasil ✓
98	Kue Lapis	20 cm	90°	1 kue	Terpotong	 Berhasil ✓

Lanjutan Tabel 4.16

No	Nama Kue	Jarak (cm)	Kemiringan (Derajat)	Jumlah	Kondisi	Hasil
99	Kue Lapis	30 cm	90°	1 kue	Terpotong	 Berhasil ✓
100	Kue Lapis	15 cm	0°	2 kue	Terpotong	 Berhasil ✓
101	Kue Lapis	20 cm	0°	2 kue	Terpotong	 Berhasil ✓

Lanjutan Tabel 4.16

No	Nama Kue	Jarak (cm)	Kemiringan (Derajat)	Jumlah	Kondisi	Hasil
102	Kue Lapis	30 cm	0°	2 kue	Terpotong	 Berhasil ✓
103	Kue Lapis	15 cm	45°	2 kue	Terpotong	 Berhasil ✓
104	Kue Lapis	20 cm	45°	2 kue	Terpotong	 Berhasil ✓

Lanjutan Tabel 4.16

No	Nama Kue	Jarak (cm)	Kemiringan (Derajat)	Jumlah	Kondisi	Hasil
105	Kue Lapis	30 cm	45°	2 kue	Terpotong	 Berhasil ✓
106	Kue Lapis	15 cm	90°	2 kue	Terpotong	 Berhasil ✓
107	Kue Lapis	20 cm	90°	2 kue	Terpotong	 Berhasil ✓

Lanjutan Tabel 4.16

No	Nama Kue	Jarak (cm)	Kemiringan (Derajat)	Jumlah	Kondisi	Hasil
108	Kue Lapis	30 cm	90°	2 kue	Terpotong	 Berhasil ✓
109	Kue Onde-Onde	15 cm	0°	1 kue	Utuh	 Berhasil ✓
110	Kue Onde-Onde	20 cm	0°	1 kue	Utuh	 Berhasil ✓

Lanjutan Tabel 4.16

No	Nama Kue	Jarak (cm)	Kemiringan (Derajat)	Jumlah	Kondisi	Hasil
111	Kue Onde-Onde	30 cm	0°	1 kue	Utuh	 Gagal ×
112	Kue Onde-Onde	15 cm	45°	1 kue	Utuh	 Berhasil ✓
113	Kue Onde-Onde	20 cm	45°	1 kue	Utuh	 Berhasil ✓

Lanjutan Tabel 4.16

No	Nama Kue	Jarak (cm)	Kemiringan (Derajat)	Jumlah	Kondisi	Hasil
114	Kue Onde-Onde	30 cm	45°	1 kue	Utuh	 Berhasil ✓
115	Kue Onde-Onde	15 cm	90°	1 kue	Utuh	 Berhasil ✓
116	Kue Onde-Onde	20 cm	90°	1 kue	Utuh	 Berhasil ✓

Lanjutan Tabel 4.16

No	Nama Kue	Jarak (cm)	Kemiringan (Derajat)	Jumlah	Kondisi	Hasil
117	Kue Onde-Onde	30 cm	90°	1 kue	Utuh	 Gagal ×
118	Kue Onde-Onde	15 cm	0°	6 kue	Utuh	 Berhasil ✓
119	Kue Onde-Onde	20 cm	0°	6 kue	Utuh	 Berhasil ✓

Lanjutan Tabel 4.16

No	Nama Kue	Jarak (cm)	Kemiringan (Derajat)	Jumlah	Kondisi	Hasil
120	Kue Onde-Onde	30 cm	0°	6 kue	Utuh	 Berhasil ✓
121	Kue Onde-Onde	15 cm	45°	6 kue	Utuh	 Berhasil ✓
122	Kue Onde-Onde	20 cm	45°	6 kue	Utuh	 Berhasil ✓

Lanjutan Tabel 4.16

No	Nama Kue	Jarak (cm)	Kemiringan (Derajat)	Jumlah	Kondisi	Hasil
123	Kue Onde-Onde	30 cm	45°	6 kue	Utuh	 Berhasil ✓
124	Kue Onde-Onde	15 cm	90°	6 kue	Utuh	 Berhasil ✓
125	Kue Onde-Onde	20 cm	90°	6 kue	Utuh	 Berhasil ✓

Lanjutan Tabel 4.16

No	Nama Kue	Jarak (cm)	Kemiringan (Derajat)	Jumlah	Kondisi	Hasil
126	Kue Onde-Onde	30 cm	90°	6 kue	Utuh	 Berhasil ✓
127	Kue Onde-Onde	15 cm	0°	1 kue	Terbuka	 Berhasil ✓
128	Kue Onde-Onde	20 cm	0°	1 kue	Terbuka	 Gagal ×

Lanjutan Tabel 4.16

No	Nama Kue	Jarak (cm)	Kemiringan (Derajat)	Jumlah	Kondisi	Hasil
129	Kue Onde-Onde	30 cm	0°	1 kue	Terbuka	 Gagal ×
130	Kue Onde-Onde	15 cm	45°	1 kue	Terbuka	 Berhasil ✓
131	Kue Onde-Onde	20 cm	45°	1 kue	Terbuka	 Berhasil ✓

Lanjutan Tabel 4.16

No	Nama Kue	Jarak (cm)	Kemiringan (Derajat)	Jumlah	Kondisi	Hasil
132	Kue Onde-Onde	30 cm	45°	1 kue	Terbuka	 Berhasil ✓
133	Kue Onde-Onde	15 cm	90°	1 kue	Terbuka	 Berhasil ✓
134	Kue Onde-Onde	20 cm	90°	1 kue	Terbuka	 Berhasil ✓

Lanjutan Tabel 4.16

No	Nama Kue	Jarak (cm)	Kemiringan (Derajat)	Jumlah	Kondisi	Hasil
135	Kue Onde-Onde	30 cm	90°	1 kue	Terbuka	 Berhasil ✓
136	Kue Onde-Onde	15 cm	0°	2 kue	Terbuka	 Berhasil ✓
137	Kue Onde-Onde	20 cm	0°	2 kue	Terbuka	 Berhasil ✓

Lanjutan Tabel 4.16

No	Nama Kue	Jarak (cm)	Kemiringan (Derajat)	Jumlah	Kondisi	Hasil
138	Kue Onde-Onde	30 cm	0°	2 kue	Terbuka	 Gagal ×
139	Kue Onde-Onde	15 cm	45°	2 kue	Terbuka	 Berhasil ✓
140	Kue Onde-Onde	20 cm	45°	2 kue	Terbuka	 Berhasil ✓

Lanjutan Tabel 4.16

No	Nama Kue	Jarak (cm)	Kemiringan (Derajat)	Jumlah	Kondisi	Hasil
141	Kue Onde-Onde	30 cm	45°	2 kue	Terbuka	 Gagal ×
142	Kue Onde-Onde	15 cm	90°	2 kue	Terbuka	 Berhasil ✓
143	Kue Onde-Onde	20 cm	90°	2 kue	Terbuka	 Berhasil ✓

Lanjutan Tabel 4.16

No	Nama Kue	Jarak (cm)	Kemiringan (Derajat)	Jumlah	Kondisi	Hasil
143	Kue Onde-Onde	20 cm	90°	2 kue	Terbuka	 Berhasil ✓
144	Kue Onde-Onde	30 cm	90°	2 kue	Terbuka	 Gagal ×
145	Kue Taripang	15 cm	0°	1 kue	Utuh	 Berhasil ✓

Lanjutan Tabel 4.16

No	Nama Kue	Jarak (cm)	Kemiringan (Derajat)	Jumlah	Kondisi	Hasil
146	Kue Taripang	20 cm	0°	1 kue	Utuh	 Gagal ×
147	Kue Taripang	30 cm	0°	1 kue	Utuh	 Gagal ×
148	Kue Taripang	15 cm	45°	1 kue	Utuh	 Berhasil ✓

Lanjutan Tabel 4.16

No	Nama Kue	Jarak (cm)	Kemiringan (Derajat)	Jumlah	Kondisi	Hasil
149	Kue Taripang	20 cm	45°	1 kue	Utuh	 Berhasil ✓
150	Kue Taripang	30 cm	45°	1 kue	Utuh	 Berhasil ✓
151	Kue Taripang	15 cm	90°	1 kue	Utuh	 Berhasil ✓

Lanjutan Tabel 4.16

No	Nama Kue	Jarak (cm)	Kemiringan (Derajat)	Jumlah	Kondisi	Hasil
152	Kue Taripang	20 cm	90°	1 kue	Utuh	 Berhasil ✓
153	Kue Taripang	30 cm	90°	1 kue	Utuh	 Berhasil ✓
154	Kue Taripang	15 cm	0°	2 kue	Utuh	 Berhasil ✓

Lanjutan Tabel 4.16

No	Nama Kue	Jarak (cm)	Kemiringan (Derajat)	Jumlah	Kondisi	Hasil
155	Kue Taripang	20 cm	0°	2 kue	Utuh	 Gagal ×
156	Kue Taripang	30 cm	0°	2 kue	Utuh	 Gagal ×
157	Kue Taripang	15 cm	45°	2 kue	Utuh	 Berhasil ✓

Lanjutan Tabel 4.16

No	Nama Kue	Jarak (cm)	Kemiringan (Derajat)	Jumlah	Kondisi	Hasil
158	Kue Taripang	20 cm	45°	2 kue	Utuh	 Berhasil ✓
159	Kue Taripang	30 cm	45°	2 kue	Utuh	 Berhasil ✓
160	Kue Taripang	15 cm	90°	2 kue	Utuh	 Berhasil ✓

Lanjutan Tabel 4.16

No	Nama Kue	Jarak (cm)	Kemiringan (Derajat)	Jumlah	Kondisi	Hasil
161	Kue Taripang	20 cm	90°	2 kue	Utuh	 Berhasil ✓
162	Kue Taripang	30 cm	90°	2 kue	Utuh	 Berhasil ✓
163	Kue Taripang	15 cm	0°	1 kue	Terpotong	 Gagal ×

Lanjutan Tabel 4.16

No	Nama Kue	Jarak (cm)	Kemiringan (Derajat)	Jumlah	Kondisi	Hasil
164	Kue Taripang	20 cm	0°	1 kue	Terpotong	 Gagal ×
165	Kue Taripang	30 cm	0°	1 kue	Terpotong	 Gagal ×
166	Kue Taripang	15 cm	45°	1 kue	Terpotong	 Berhasil ✓

Lanjutan Tabel 4.16

No	Nama Kue	Jarak (cm)	Kemiringan (Derajat)	Jumlah	Kondisi	Hasil
167	Kue Taripang	20 cm	45°	1 kue	Terpotong	 Berhasil ✓
168	Kue Taripang	30 cm	45°	1 kue	Terpotong	 Berhasil ✓
169	Kue Taripang	15 cm	90°	1 kue	Terpotong	 Berhasil ✓

Lanjutan Tabel 4.16

No	Nama Kue	Jarak (cm)	Kemiringan (Derajat)	Jumlah	Kondisi	Hasil
170	Kue Taripang	20 cm	90°	1 kue	Terpotong	 Berhasil ✓
171	Kue Taripang	30 cm	90°	1 kue	Terpotong	 Berhasil ✓
172	Kue Taripang	15 cm	0°	2 kue	Terpotong	 Berhasil ✓

Lanjutan Tabel 4.16

No	Nama Kue	Jarak (cm)	Kemiringan (Derajat)	Jumlah	Kondisi	Hasil
173	Kue Taripang	20 cm	0°	2 kue	Terpotong	 Gagal ×
174	Kue Taripang	30 cm	0°	2 kue	Terpotong	 Gagal ×
175	Kue Taripang	15 cm	45°	2 kue	Terpotong	 Berhasil ✓

Lanjutan Tabel 4.16

No	Nama Kue	Jarak (cm)	Kemiringan (Derajat)	Jumlah	Kondisi	Hasil
176	Kue Taripang	20 cm	45°	2 kue	Terpotong	 Berhasil ✓
177	Kue Taripang	30 cm	45°	2 kue	Terpotong	 Berhasil ✓
178	Kue Taripang	15 cm	90°	2 kue	Terpotong	 Berhasil ✓

Lanjutan Tabel 4.16

No	Nama Kue	Jarak (cm)	Kemiringan (Derajat)	Jumlah	Kondisi	Hasil
179	Kue Taripang	20 cm	90°	2 kue	Terpotong	 Berhasil ✓
180	Kue Taripang	30 cm	90°	2 kue	Terpotong	 Berhasil ✓

Berdasarkan data yang disajikan pada Tabel 4.16, dilakukan analisis mendalam terhadap performa model *Convolutional Neural Network* (CNN) dalam mengklasifikasikan berbagai jenis kue tradisional berdasarkan variasi jarak, sudut / Kemiringan (Derajat) pengambil gambar, kondisi kue, dan jumlah kue.

a. Pengaruh Jarak Pengambilan Gambar (15cm, 20cm, 30cm)

Berikut adalah analisis akurasi berdasarkan jarak pengambilan gambar **(**15cm, 20cm, 30cm**)** yang disajikan dalam Tabel 4.17.

Tabel 4.17 Tabel Akurasi berdasarkan Jarak Pengambilan Gambar

Nama Kue	Jarak (cm)	Prediksi Benar	Prediksi Salah	Akurasi
Kue Barongko	15	12	0	100%
	20	12	0	100%
	30	12	0	100%
Kue Dange	15	12	0	100%
	20	10	2	83,33%
	30	9	3	75,00%
Kue Lapis	15	12	0	100%
	20	12	0	100%
	30	10	2	83,33%
Kue Onde-Onde	15	12	0	100%
	20	11	1	91,67%
	30	6	6	50,00%
Kue Taripang	15	11	1	91,67%
	20	8	4	66,67%
	30	8	4	66,67%

Jarak antara kamera dan objek secara umum menunjukkan korelasi negatif dengan tingkat keyakinan model.

Jarak yang lebih dekat (15 cm dan 20 cm) cenderung menghasilkan skor keyakinan yang lebih tinggi. Pada jarak ini, objek kue mengisi sebagian besar frame gambar, memungkinkan CNN untuk mengekstraksi fitur dengan lebih detail.

Seiring bertambahnya jarak ke 30 cm, terdapat beberapa kasus di mana akurasi menurun atau terjadi kegagalan (misal, No. 3, 84, 93, 111 pada Table 4.16). Hal ini dapat diatribusikan pada berkurangnya resolusi efektif objek dalam gambar, sehingga beberapa fitur kunci menjadi kurang jelas bagi model.

b. Pengaruh Sudut Pengambilan Gambar (0°, 45°, 90°)

Berikut adalah analisis akurasi berdasarkan sudut pengambilan gambar (0°, 45°, 90°) yang disajikan dalam Tabel 4.18.

Tabel 4.18 Tabel Akurasi berdasarkan Sudut Pengambilan Gambar

Nama Kue	Kemiringan (Derajat) / Sudut Pengambilan Gambar	Prediksi Benar	Prediksi Salah	Akurasi (%)
Kue Barongko	0°	12	0	100%
	45°	12	0	100%
	90°	12	0	100%
Kue Dange	0°	10	2	83,33%
	45°	11	1	91,67%
	90°	10	2	83,33%
Kue Lapis	0°	10	2	83,33%
	45°	12	0	100%
	90°	12	0	100%
Kue Onde-Onde	0°	8	4	58,33%
	45°	11	1	91,67%
	90°	10	2	83,33%
Kue Taripang	0°	3	9	25,00%
	45°	12	0	100%
	90°	12	0	100%

Sudut pengambilan gambar menunjukkan pengaruh yang bervariasi tergantung pada jenis kuenya.

Faktor ini menjadi sangat krusial untuk membedakan kue dengan bentuk dan warna yang mirip, seperti Kue Taripang dan Kue Dange. Pada sudut 0° (tampak samping), model konsisten salah mengklasifikasikan Kue Taripang sebagai Kue Dange (misal, No. 146, 147, 163-165 pada Tabel 4.16). Namun, pada sudut 45° dan 90°, di mana profil tiga dimensi kue lebih terlihat, model berhasil mengidentifikasi Kue Taripang dengan benar dan dengan keyakinan tinggi (misal, No. 148, 149, 166-171 pada Tabel 4.16).

Untuk kue dengan fitur yang sangat jelas dari berbagai sudut seperti Kue Lapis (pola lapisan) dan Kue Barongko (bungkus khas), perubahan sudut tidak memberikan dampak negatif yang signifikan terhadap akurasi.

c. Pengaruh Jumlah Kue (1 kue vs. lebih dari 1)

Berikut adalah analisis akurasi berdasarkan jumlah kue (1 kue vs. lebih dari 1) yang disajikan dalam Tabel 4.19.

Tabel 4.19 Tabel Akurasi berdasarkan Jumlah Kue

Nama Kue	Jumlah Kue	Prediksi Benar	Prediksi Salah	Akurasi (%)
Kue Barongko	1	18	0	100%
	Lebih dari 1	18	0	100%
Kue Dange	1	15	3	83,33%
	Lebih dari 1	16	2	88,89%
Kue Lapis	1	17	1	94,44%
	Lebih dari 1	17	1	94,44%
Kue Onde-Onde	1	14	4	77,78%
	Lebih dari 1	14	3	82,35%
Kue Taripang	1	13	5	72.22%
	Lebih dari 1	14	4	77,78%

Berdasarkan data yang ada, jumlah objek (satu atau banyak kue) tidak tampak menjadi faktor penentu utama yang menurunkan performa model. Model terlihat mampu menangani deteksi pada gambar yang berisi dua kue atau lebih, dengan hasil yang sebanding dengan deteksi pada satu kue, asalkan faktor-faktor lain (kondisi, sudut, jarak) bersifat ideal.

d. Pengaruh Kondisi Kue (Utuh vs. Terpotong/Terbuka)

Berikut adalah analisis akurasi berdasarkan kondisi kue (Utuh / Utuh vs. terpotong / terbuka) yang disajikan dalam Tabel 4.16.

Tabel 4.20 Tabel Akurasi berdasarkan Kondisi Kue

Nama Kue	Kondisi Kue	Prediksi Benar	Prediksi Salah	Akurasi (%)
Kue Barongko	Utuh/Utuh	18	0	100%
	Terpotong/Terbuka	18	0	100%
Kue Dange	Utuh/Utuh	18	0	100%
	Terpotong/Terbuka	13	5	72,22%
Kue Lapis	Utuh/Utuh	17	1	94,44%
	Terpotong/Terbuka	17	1	94,44%
Kue Onde-Onde	Utuh/Utuh	15	3	83,33%
	Terpotong/Terbuka	13	5	72,22%
Kue Taripang	Utuh/Utuh	14	4	77,78%
	Terpotong/Terbuka	13	5	72,22%

Kondisi fisik kue menjadi faktor paling signifikan yang memengaruhi akurasi deteksi.

Secara umum, model menunjukkan performa yang sangat baik pada kue dalam kondisi Utuh (`Utuh`). Kue seperti Kue Barongko dan Kue Lapis berhasil diidentifikasi dengan skor keyakinan (confidence score) yang konsisten tinggi, seringkali di atas 95%, di hampir semua skenario pengujian. Hal ini mengindikasikan bahwa model telah berhasil mempelajari fitur-fitur eksternal yang unik dari kue-kue tersebut, seperti bungkusan daun pisang pada Kue Barongko dan pola berlapis pada Kue Lapis.

Performa model menurun secara drastis ketika dihadapkan pada kue yang telah dipotong (`Terpotong`/`Terbuka`). Hal ini sangat terlihat pada Kue Dange, Kue Onde-Onde, dan Kue Taripang.

Pada kasus Kue Onde-Onde Terbuka (misal, No. 128, 129, 137, 138 pada Tabel 4.16). model seringkali gagal dan salah mengklasifikasikannya sebagai Kue Dange atau Kue Taripang. Karena tekstur internal (isian) tidak memiliki fitur

distingtif seperti kulit luar yang bertabur parut kelapa, sehingga membingungkan model.

Demikian pula pada Kue Dange dan Kue Taripang yang terpotong, model menunjukkan kecenderungan salah klasifikasi, terutama pada jarak yang lebih jauh atau sudut yang kurang optimal.

Dapat disimpulkan bahwa model lebih andal dalam mengenali fitur eksterior (bentuk, warna, tekstur permukaan) daripada fitur interior kue.

Untuk mengidentifikasi titik kegagalan model, pengujian lebih lanjut dilakukan pada skenario yang lebih sulit, seperti jarak yang lebih jauh dan kondisi pencahayaan yang beragam.

e. Pengujian Jarak Tetap (50cm)

Pengujian ini dirancang sebagai stress test untuk jarak. Hasil Pengujian dapat dilihat pada tabel 4.21.

Tabel 4.21 Tabel Pengujian Fitur Deteksi dengan Jarak 50cm

No	Nama Kue	Jarak 50 cm		
		0	45	90
1	Kue Barongko	 Berhasil ✓	 Berhasil ✓	 Berhasil ✓

Lanjutan Tabel 4.21

No	Nama Kue	Jarak 50 cm		
		0	45	90
2	Kue Dange	 Berhasil ✓	 Berhasil ✓	 Berhasil ✓
3	Kue Lapis	 Gagal ×	 Gagal ×	 Gagal ×
4	Kue Onde-Onde	 Gagal ×	 Gagal ×	 Gagal ×

Lanjutan Tabel 4.17

No	Nama Kue	Jarak 50 cm		
		0	45	90
5	Kue Taripang	 Gagal ×	 Gagal ×	 Gagal ×

Berdasarkan hasil pengujian pada Tabel 4.21, evaluasi robustisitas model Convolutional Neural Network (CNN) pada jarak pengambilan gambar 50 cm dengan variasi sudut 0°, 45°, dan 90° menunjukkan adanya perbedaan performa yang signifikan antar jenis kue. Analisis ini mengungkapkan bahwa faktor jarak menjadi variabel paling kritis yang membatasi kemampuan operasional model. Keberhasilan model dalam mengidentifikasi Kue Barongko secara konsisten di semua sudut, bahkan dengan tingkat keyakinan mencapai 99,88%, diatribusikan pada fitur makro yang unik dan dominan, yaitu bentuk limas dari bungkusan daun pisangnya yang tetap dapat dikenali dari jauh. Demikian pula, Kue Dange berhasil diklasifikasikan dengan benar, meskipun dengan tingkat keyakinan yang lebih rendah (berkisar antara 52,1% hingga 77,56%), yang mengindikasikan bahwa bentuk solid dan warnanya yang homogen cukup distingtif, namun hilangnya detail tekstur akibat jarak menurunkan keyakinan prediksi model.

Sebaliknya, model mengalami kegagalan klasifikasi secara konsisten pada Kue Lapis, Kue Onde-Onde, dan Kue Taripang. Pada jarak 50 cm, fitur-fitur esensial yang menjadi pembeda utama—seperti pola berlapis pada Kue Lapis atau parutan kelapa pada Kue Onde-Onde—mengalami degradasi visual yang parah sehingga menjadi terlalu kecil dan kabur untuk diekstraksi oleh model. Akibatnya, model cenderung salah mengklasifikasikannya sebagai Kue Dange atau Kue Barongko, yang kemungkinan memiliki representasi fitur visual yang lebih kuat dalam dataset pelatihan. Dalam skenario jarak jauh ini, pengaruh Kemiringan (Derajat) sudut menjadi sekunder. Meskipun sudut dapat memperlihatkan profil tiga dimensi objek, manfaat ini menjadi tidak relevan ketika fitur-fitur krusial sudah hilang karena jarak. Hal ini menegaskan bahwa model memiliki rentang jarak operasional efektif yang terbatas, dan untuk objek yang identifikasinya bergantung pada detail beresolusi tinggi, perubahan sudut tidak mampu mengompensasi hilangnya informasi visual yang disebabkan oleh jarak.

f. Pengujian Jenis Sumber Cahaya

Pengujian terakhir mengevaluasi dampak berbagai kondisi pencahayaan terhadap akurasi model. Parameter tetap yang digunakan adalah jarak 20cm dengan sudut pengambilan gambar 45°.

Tabel 4.22 Tabel Pengujian Fitur Deteksi dengan Variable Cahaya

No	Nama Kue	Pencahayaan Lampu Ruangan	Pencahayaan Lampu Belajar	Pencahayaan dari samping	Pencahayaan dari Flash HP	Hasil
1	Kue Barongko	✓	×	×	×	 Berhasil ✓
2	Kue Barongko	×	✓	×	×	 Berhasil ✓

Lanjutan Tabel 4.22

No	Nama Kue	Pencahayaan Lampu Ruangan	Pencahayaan Lampu Belajar	Pencahayaan dari samping	Pencahayaan dari Flash HP	Hasil
3	Kue Barongko	×	×	✓	×	 Berhasil ✓
4	Kue Barongko	×	×	×	✓	 Berhasil ✓
5	Kue Dange	✓	×	×	×	 Berhasil ✓

Lanjutan Tabel 4.22

No	Nama Kue	Pencahayaan Lampu Ruangan	Pencahayaan Lampu Belajar	Pencahayaan dari samping	Pencahayaan dari Flash HP	Hasil
6	Kue Dange	×	✓	×	×	 Berhasil ✓
7	Kue Dange	×	×	✓	×	 Berhasil ✓
8	Kue Dange	×	×	×	✓	 Berhasil ✓

Lanjutan Tabel 4.22

No	Nama Kue	Pencahayaan Lampu Ruangan	Pencahayaan Lampu Belajar	Pencahayaan dari samping	Pencahayaan dari Flash HP	Hasil
9	Kue Lapis	✓	×	×	×	 Gagal ×
10	Kue Lapis	×	✓	×	×	 Gagal ×
11	Kue Lapis	×	×	✓	×	 Gagal ×

Lanjutan Tabel 4.22

No	Nama Kue	Pencahayaan Lampu Ruangan	Pencahayaan Lampu Belajar	Pencahayaan dari samping	Pencahayaan dari Flash HP	Hasil
12	Kue Lapis	×	×	×	✓	 Berhasil ✓
13	Kue Onde-Onde	✓	×	×	×	 Berhasil ✓
14	Kue Onde-Onde	×	✓	×	×	 Gagal ×

Lanjutan Tabel 4.22

No	Nama Kue	Pencahayaan Lampu Ruangan	Pencahayaan Lampu Belajar	Pencahayaan dari samping	Pencahayaan dari Flash HP	Hasil
15	Kue Onde-Onde	×	×	✓	×	 Berhasil ✓
16	Kue Onde-Onde	×	×	×	✓	 Berhasil ✓
17	Kue Taripang	✓	×	×	×	 Berhasil ✓

Lanjutan Tabel 4.22

No	Nama Kue	Pencahayaan Lampu Ruangan	Pencahayaan Lampu Belajar	Pencahayaan dari samping	Pencahayaan dari Flash HP	Hasil
18	Kue Taripang	×	✓	×	×	 Berhasil ✓
19	Kue Taripang	×	×	✓	×	 Gagal ×
20	Kue Taripang	×	×	×	✓	 Berhasil ✓

Berikut adalah analisis akurasi berdasarkan skenario pencahayaan yang disajikan dalam Tabel 4.23.

Tabel 4.23 Akurasi Berdasarkan Jumlah Kue

Skenario Pencahayaan	Prediksi Benar	Prediksi Salah	Akurasi
Lampu Ruangan	3	2	60%
Lampu Belajar	3	2	60%
Pencahayaan dari Samping	4	1	80%
Pencahayaan dari Flash HP	5	0	100%

Pengujian menunjukkan bahwa kondisi pencahayaan secara signifikan memengaruhi performa model CNN. Kue dengan fitur visual yang kuat dan berbasis bentuk, seperti Kue Barongko, dapat dikenali dengan akurasi tinggi di hampir semua kondisi cahaya. Sebaliknya, kue yang identifikasinya bergantung pada fitur detail seperti warna berlapis (Kue Lapis) atau tekstur permukaan (Kue Onde-Onde) sangat rentan terhadap pencahayaan yang tidak optimal. Secara konsisten, penggunaan cahaya langsung dan terang dari *flash* HP memberikan hasil paling andal, sementara cahaya ambien dari lampu ruangan atau lampu belajar yang terfokus seringkali menyebabkan penurunan akurasi atau kegagalan klasifikasi. Hal ini membuktikan bahwa kualitas dan arah cahaya merupakan faktor eksternal kritis yang menentukan keberhasilan deteksi kue.

J. Detail Aplikasi

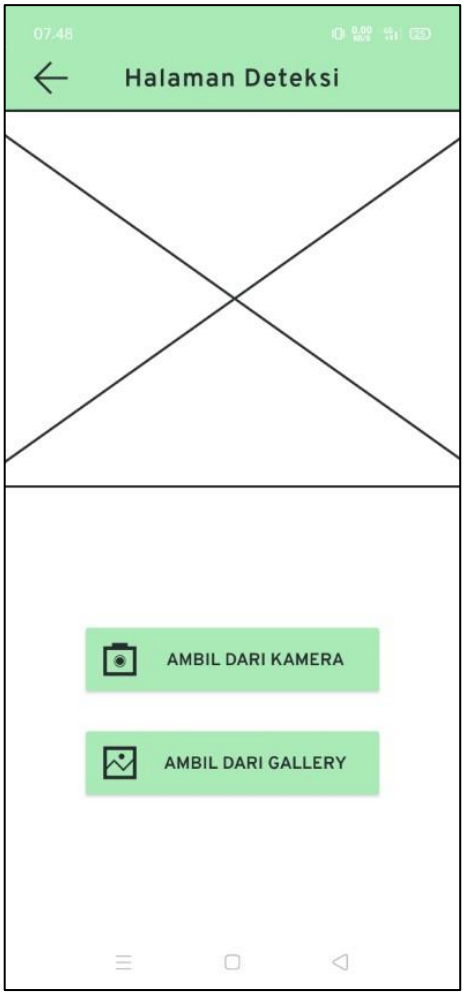
1. Halaman Depan



Gambar 4.27 Antarmuka Halaman Depan

Halaman depan atau halaman pertama yang ada pada aplikasi menampilkan semua fitur yang ada pada aplikasi yaitu, deteksi kue, katalog kue, tentang aplikasi, dan keluar.

2. Halaman Deteksi Kue



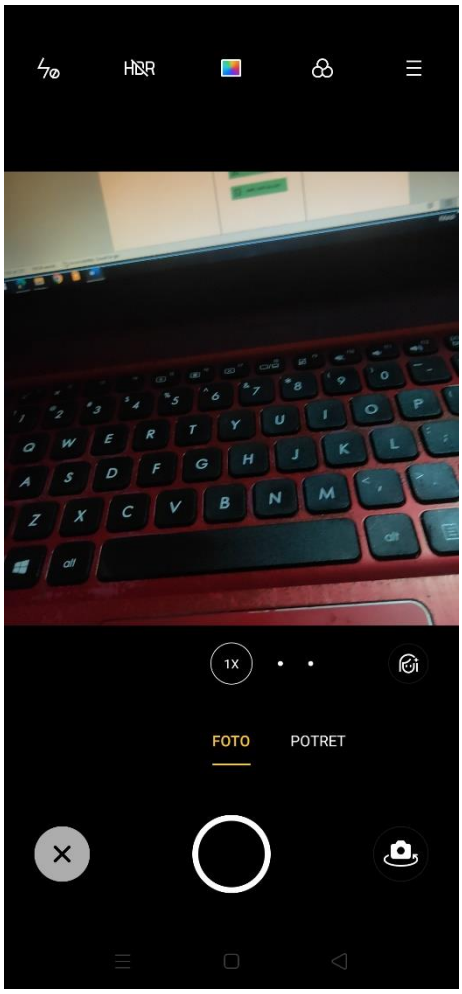
Gambar 4.28 Antarmuka Halaman Deteksi Kue

Halaman deteksi kue adalah fitur halaman utama pada aplikasi yang berfungsi untuk menginput gambar dengan memiliki opsi memilih gambar dari galeri atau mengambil gambar secara langsung.

Pada aplikasi ini terdapat, pengguna diberikan dua opsi untuk mengunggah gambar: melalui kamera atau memilih gambar dari galeri. Kedua opsi ini menawarkan cara yang berbeda untuk mendapatkan gambar. Opsi pertama memungkinkan pengguna mengambil gambar langsung menggunakan kamera

perangkat, sementara opsi kedua memungkinkan pengguna memilih gambar yang sudah tersimpan di galeri.

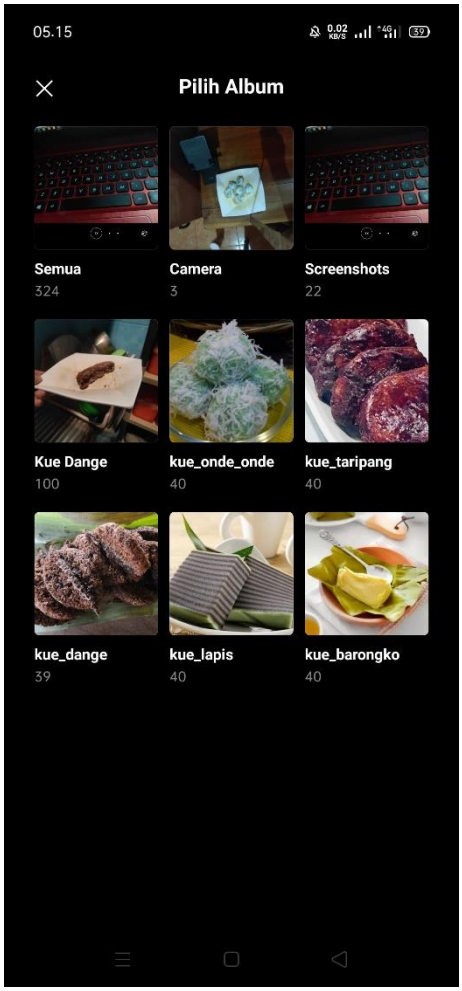
3. Halaman Ambil Gambar



Gambar 4.29 Antarmuka Halaman Ambil Kue

Halaman ambil gambar merupakan fitur yang digunakan untuk menginput gambar pengenalan dengan mengambil gambar secara langsung.

4. Halaman Ambil Galeri



Gambar 4.30 Antarmuka Halaman Ambil Galeri

Halaman ambil galeri adalah fitur yang digunakan untuk menginput gambar yang terdapat pada galeri handphone yang dimiliki.

5. Halaman Katalog Kue



Gambar 4.31 Antarmuka Halaman Katalog Kue

Halaman katalog kue adalah halaman yang berisi daftar kue atau katalog kue disertai dengan deskripsi singkatnya.

6. Halaman Deskripsi Kue



Gambar 4.32 Antarmuka Halaman Deskripsi Kue

Halaman deskripsi kue adalah halaman yang terdapat informasi kue-kue yang terdiri dari penjelasan kue, resep, dan cara pembuatan.

7. Halaman Tentang Aplikasi



Gambar 4.33 Antarmuka Halaman Tentang Aplikasi

Halaman tentang aplikasi adalah halaman yang memiliki informasi terkait aplikasi pengenalan kue tradisional.

BAB V

PENUTUP

A. Kesimpulan

Sejalan dengan pengkajian yang sudah dijabarkan dalam BAB IV, dalam pembuatan Aplikasi Pengenalan Kue Tradisional Bugis menggunakan metode *Convolutional Neural Network* (CNN) berbasis Android dapat ditarik kesimpulan sebagai berikut:

1. Aplikasi ini merupakan aplikasi berbasis Android yang dibuat menggunakan Android Studio dengan bahasa pemrograman Java, serta mengimplementasikan TensorFlow Lite sebagai alat bantu dalam proses pengenalan gambar menggunakan metode Convolutional Neural Network (CNN).
2. Hasil pengujian data testing menunjukkan Model EfficientNetB0 batch size 64 dan epoch 50 sebagai model dengan performa terbaik. Model ini berhasil mencapai akurasi sebesar 97.00% pada set validasi dan akurasi final sebesar 92.40% pada set pengujian (*test set*). Hasil ini menunjukkan kemampuan model untuk menggeneralisasi dengan baik dan menjadikannya kandidat yang ideal untuk diimplementasikan pada perangkat Android.
3. Hasil pengujian fitur deteksi aplikasi menggunakan kamera menujukkan performa optimal dalam rentang jarak 15 cm hingga 30 cm dan mampu mengatasi variasi sudut pandang hingga 90° untuk kue dengan fitur visual yang jelas. Namun, performa model menurun signifikan pada jarak 50 cm, di mana

fitur-fitur detail kue menjadi kabur, dan pada kue dalam kondisi terpotong yang kehilangan fitur eksternal khasnya. Selain itu, kondisi pencahayaan juga terbukti sangat memengaruhi performa model. Cahaya langsung dari flash HP konsisten memberikan hasil terbaik, sementara pencahayaan ambien sering menyebabkan penurunan akurasi atau salah klasifikasi.

B. Saran

Sehubungan dengan kesimpulan tersebut, beberapa saran dapat diajukan sebagai berikut:

1. Melakukan penambahan jumlah class dan jumlah dataset gambar untuk meningkatkan test akurasi model terhadap kue tradisional bugis.
2. Untuk ke-depannya fokus pada pengembangan dan peningkatan fitur aplikasi berdasarkan umpan balik pengguna dan penambahan jumlah kue yang dapat di deteksi.

DAFTAR PUSTAKA

Abdiansyah, Baharuddin, & Said, M. S. (2024). Klasifikasi jenis beras menggunakan metode convolutional neural network pada arsitektur MobileNet. *Sistem Informasi dan Teknik Komputer*. 9(2): 299-305.

Adi, R. P., Koswara, Y., Tashika, J., Devi, Y., & Saifudin, A. (2020). Pengujian black box pada aplikasi minimarket menggunakan metode equivalence partitioning. *Jurnal Teknologi Sistem Informasi dan Aplikasi*. 3(2): 100-106.

Andriyanti, A., & Setiawan, B. (2024). Upaya pelestarian dan pengembangan kue tradisional Betawi. *Jurnal Pariwisata dan Perhotelan*. 2(1): 1-14.

Anisa, Andriani, D., & Achmadi, N. S. (2022). Pengenalan kue basah tradisional Suku Bugis berbungkus daun pisang sebagai sajian kebudayaan Kabupaten Soppeng Sulawesi Selatan. *Hospitality and Gastronomy Research Journal*. 4(1): 238-257.

Arifianto, R. (2014). *Materi Flowchart*. Diperoleh dari: https://rahmatarifianto.wordpress.com/2014/11/20/pengertian-Flowchart-dan-jenis-jenisnya.html. (Diakses 11 Mei 2024).

Black, R. (2022). *Software testing techniques*. Indianapolis : Pearson Education, inc.

Daqiqi, I. (2021). *Machine Learning : Teori, studi kasus dan implementasi menggunakan Python*. Riau : UR PRESS.

Fukushima, K. (1980). Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*. 36(4): 193–202.

Ginting, Y. M., Yulastri, A., & Syarif, W. (2017). Inventatisasi Jenis dan Resep Kue-Kue Tradisional di Kota Padang, Padang : UNP Vol 14(1).

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.

Gusdevi, H., Kuswayati, S., Iqbal, M., Abu Bakar, M. F., Novianti, N., & Ramadan, R., (2022), Pengujian white-box pada aplikasi Debt Manager berbasis Android. *Jurnal NARATIF* 4(1).

Haridas, R., & Jyothi, R. L. (2019). Convolutional neural networks: A comprehensive survey. *International Journal of Computer Engineering and Technology*. 10(2): 1–12.

Hasnawati, & Asriadi. (2022). Aplikasi Game Memasak Kue Tradisional Bugis Menggunakan Construct 2 Berbasis Android. *Jurnal Sintaks Logika*. 2(2): 55-60.

Haykin, S. (2009). Neural networks: A comprehensive foundation. Prentice Hall.

Hestiningsih, I., Thohari, A. N. A., Kurnianingsih, & Kamarudin, N. D. (2023). Mobile Skin Disease Classification using MobileNetV2 and NASNetMobile. *International Journal on Advanced Science, Engineering and Information Technology*. 13(4): 1472-1479.

Howard, A., Sandler, M., Chu, G., Chen, L.-C., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., Vasudevan, V., Le, Q. V., & Adam, H. (2019). *Searching for MobileNetV3*. Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). 1314–1324.

Ismah. (2017). *Pemrograman Komputer Dasar Dasar Python*. Jakarta Utara: Fakultas Ilmu Komputer UMJ.

I Wayan Suartika, E. P., Wijaya, A. Y., & Soelaiman, R. (2016). Klasifikasi citra menggunakan Convolutional Neural Network (CNN) pada Caltech 101. *Jurnal Teknik ITS*. 5(1).

Kapojos, S. M., & Wijaya, H. (2018). Mengenal Budaya Suku Bugis Pendekatan Misi Terhadap Suku Bugis. *Lembaga STANKN Kupang*. 6(2).

Khairunisa, Y., Mardiani, E., Rahmansyah, N., Ambarwari, A., Yuspita, Z., Alia, P. A., Santoso, P. B., Hayati, N., Kartiko, E. Y., Prayogo, J. S., Setyadi, A. T., Matondang, N., & Sayekti, I. H. (2023). *Dasar-Dasar Pemrograman*. Yogyakarta: PT Penamuda Media.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 25.

LeCun, Y. (1990). Handwritten digit recognition with a back-propagation network [Technical report]. AT&T Bell Laboratories.

Lubis, A. (2016). Basis Data Dasar Edisi Pertama. Yogyakarta: Deepublish.

Murtiwiyati, & Lauren, G. (2013). Rancang bangun aplikasi pembelajaran budaya Indonesia untuk anak sekolah dasar berbasis Android. *Jurnal Ilmiah Komputasi*.

Nahriana, & Tawani, R. (2022). Diversity of Bugis-Makassar Ethnic Food in South Sulawesi. IJCHT, 2022, 1-7.

Nugraha, A. A., & Setyati, E. (2021). Klasifikasi aktivitas manusia pada video menggunakan Convolutional Neural Network (CNN). *Jurnal Teknik ITS*. 10(2): A365-A370.

Nur, A. M., Djirong, A., & Arifin, I. (2022). Studi Etnomatematika pada Kue Tradisional Suku Bugis. *ELIPS: Jurnal Pendidikan Matematika, 5*(1): 121-127.

Sakhavi, S., Guan, C., & Yan, S. (2018). Learning temporal information for brain-computer interface using convolutional neural networks. IEEE Transactions on Neural Networks and Learning Systems. 29(11): 5619–5629.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.-C. (2018). *MobileNetV2: Inverted Residuals and Linear Bottlenecks*. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 4510–4520.

Satyaputra, A., & Aritonang, E, M. (2016). *Let's build your Android apps with Android Studio*. Jakarta: PT Elex Media Komputindo.

Seafudin, A. (2022). Pengujian perangkat lunak (Edisi 2). Bandung: Informatika Bandung.

Shadiq, J., Safei A., & Loly., R.W.R. (2021). Pengujian Aplikasi Peminjaman Kendaraan Operasional Kantor Menggunakan BlackBox Testing. *Information Management for Educators and Professional*. 5(2).

Shoshana, M. G., & Santoso J. (2014). Pengaruh Budaya, Sub Budaya, Kelas Sosial, Dan Persepsi Kualitas Terhadap Perilaku Keputusan Pembelian Kue Tradisional Oleh Mahasiswa Di Surabaya. *Surabaya : Universitas Kristen Petra*. 2(1).

Sora, N. (2015). *Pengertian UML dan Jenis - jenisnya*. Diperoleh dari: http://www. pengertianku.net/2015/09/pengertian-uml-dan-jenis jenisnya-serta-contoh-diagramnya.html. (Diakses 11 Mei 2023).

Sudirman, Masnur, & Yasin, S. (2021). Rancang Bangun Aplikasi Ensiklopedia Makanan Tradisional Bugis Beserta Cara Pembuatannya Berbasis Android. *Sintaks Logika*. 1(2): 62-65.

Sumarlie, D., Lubis, C., & Handhayani, T. (2022). Pengenalan Kue Tradisional Indonesia Menggunakan Algoritma Convolutional Neural Network. *Computatio: Journal of Computer Science and Information Systems*. 6(2): 164-171.

Syahrial, Badollahi, M. Z., & Winowatan, W. J. (2022). *Implementasi kudapan tradisional Bugis pada industri perhotelan*. Makassar: Poltekpar Press.

Tan, M., & Le, Q. V. (2019). *EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks*. Proceedings of the 36th International Conference on Machine Learning (ICML), 6105–6114.

Wardani, K. R. R., Endarmono, S. S., & Sholihah, M. (2022). Klasifikasi penyakit daun anggur menggunakan metode Convolutional Neural Network. *Jurnal Tekno Insentif*. 16(2): 115-125.

Widodo, B. (2016). Machine learning and computational intelligence.

Yudhanto, Y., & Wijayanto, A. (2017). *Mudah membuat dan berbisnis aplikasi Android dengan Android Studio*. Jakarta: PT Elex Media Komputindo.

Yunus, M., & Anwar, Y. (2022). Aplikasi Penerjemah Bahasa Isyarat Indonesia ke Dalam Huruf Abjad. *Sintaks Logika*. 2(1): 257-262.