

Aplikasi Android Pendeteksi Kualitas Beras Berbasis *Machine Learning* Menggunakan Metode *Convolutional Neural Network*

Febriyanti Paramudita^{*1}, Mulki Indana Zulfa²

^{1,2}Teknik Elektro, Fakultas Teknik, Universitas Jenderal Soedirman, Indonesia
Email: ¹febriyanti.paramudita@mhs.unsoed.ac.id, ²mulki_indanazulfa@unsoed.ac.id

Abstrak

Beras merupakan makanan pokok dan bahan utama dalam berbagai produk pangan lainnya, yang memerlukan tingkat kualitas tertentu. Kualitas beras meliputi sifat fisik yang memengaruhi penampilan dan menentukan kualitasnya saat dimasak. Hal tersebut mendorong kebutuhan akan metode pendeteksian kualitas beras yang efisien. Penelitian ini menyajikan aplikasi android untuk pendeteksian kualitas beras berbasis *machine learning*. Metode pengembangan aplikasinya menggunakan agile yang memberikan fleksibilitas menangani perubahan kebutuhan dan umpan balik dari pengguna selama proses pengembangan. Lebih lanjut model *machine learning* dibuat menggunakan metode *Convolutional Neural Network* (CNN) dengan *library* TensorFlow dan Keras serta arsitektur MobileNet. Bahasa pemrograman yang digunakan adalah *python* dengan lingkungan simulasinya menggunakan Google Colab. Model yang sudah dibangun dilatih menggunakan dataset berupa 1800 gambar, 1440 diantaranya adalah *data training* dengan 25 epoch. Hasil simulasi menunjukkan training loss sebesar 0.0012, dengan nilai akurasi 99.44%.

Kata kunci: *agile, android, convolution neural network, machine learning*

Build and Design of an Android Application for Rice Quality Detection based on Machine Learning using Convolutional Neural Network (CNN)

Abstract

Rice is a staple food and the main ingredient in various other food products, which require a certain level of quality. The quality of rice includes physical properties that affect its appearance and determine its quality when cooked. This prompted the need for an efficient rice quality detection method. This study presents an android application for detecting the quality of rice based on machine learning. The application development method uses agile which provides flexibility to handle changing needs and feedback from users during the development process. Furthermore, machine learning models are created using the Convolutional Neural Network (CNN) method with the TensorFlow and Keras libraries and the MobileNet architecture. The programming language used is python with the simulation environment using Google Colab. The model that has been built is trained using a dataset in the form of 1800 images, 1440 of which are training data with 25 epochs. The simulation results show a training loss of 0.0012, with an accuracy value of 99.44%.

Keywords: *agile, android, convolution neural network, machine learning*

1. PENDAHULUAN

Beras merupakan makanan pokok dan bahan utama dalam berbagai produk pangan lainnya. Handani *et al.* (2021) menjelaskan bahwa permintaan beras akan cenderung meningkat sejalan dengan meningkatnya jumlah penduduk. Permintaan beras akan terus meningkat seiring dengan pertambahan jumlah penduduk. Permintaan beras di seluruh Indonesia terus meningkat sekitar 1,00 persen setiap tahun hingga tahun 2050 [1]. Oleh karena itu, beras yang tersedia di pasaran harus memenuhi standar dan kualitas yang baik. Mutu beras dipengaruhi oleh beberapa ciri fisik seperti (1) ukuran dan bentuk butiran, (2) tingkat derajat sosoh, (3) kejernihan, (4) kebersihan dan kemurniannya. Sebab beras dikonsumsi dalam bentuk butiran utuh, maka ciri-ciri fisik tersebut berperan penting dalam kualitasnya [2].

Kualitas beras dapat dikategorikan berdasarkan bentuk dan warna. Warna dan bentuk beras mempengaruhi kualitasnya. Semakin putih, bersih, dan utuh beras, semakin baik kualitasnya. Penilaian tingkat keputihan dan kebersihan beras didapatkan melalui analisis nilai HSV pada gambar beras, sementara penilaian tingkat keutuhan

diperoleh melalui analisis luas objek beras. Berdasarkan kualitasnya, beras dapat dikategorikan ke dalam kualitas beras premium, medium, dan tidak layak.

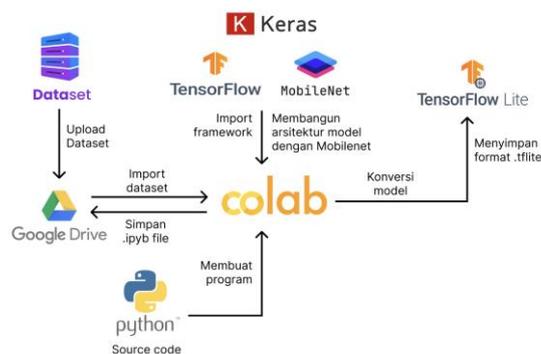
Masyarakat perlu menjadi lebih cermat dalam memeriksa kualitas beras, apakah beras tersebut cocok untuk dimasak atau tidak. Namun, kenyataannya, banyak masyarakat yang belum memiliki pengetahuan khusus tentang cara mengenali kualitas beras yang sesuai untuk dimasak [3]. Dalam mengatasi tantangan ini, pemanfaatan teknologi informasi, khususnya melalui aplikasi android dapat menjadi solusi yang efektif. Dengan memanfaatkan perangkat bergerak yang dimiliki oleh masyarakat, aplikasi android dapat menjadi alat yang mudah diakses dan praktis untuk membantu mendeteksi kualitas beras. Pemanfaatan teknologi Android sebagai platform aplikasi memberikan keunggulan dalam hal aksesibilitas dan fleksibilitas. Dengan menggunakan perangkat Android, pengguna dapat dengan mudah menginstal dan mengakses aplikasi ini melalui *smartphone* mereka, sehingga memudahkan mereka dalam melakukan pengecekan kualitas beras di mana saja dan kapan saja [4]. Keberadaan aplikasi ini memberikan dampak positif bagi masyarakat, terutama bagi petani dan konsumen beras. Petani dapat dengan cepat mengetahui kualitas beras yang dihasilkan dari hasil panen mereka.

Penelitian ini bertujuan mengembangkan sebuah aplikasi android berbasis *machine learning* yang dapat digunakan untuk mendeteksi kualitas beras. Pengembangan aplikasinya menggunakan teknologi *machine learning* sebagai inti dari pendeteksian kualitas beras. Model *Machine Learning* menggunakan metode *Convolutuional Neural Network* (CNN) dengan arsitektur MobileNet telah diimplementasikan untuk mengenali ciri-ciri fisik beras yang menentukan kualitasnya.

2. METODE PENELITIAN

2.1. Model Deteksi dengan *Machine Learning*

Penelitian ini memerlukan model *machine learning* yang dapat mengklasifikasikan kualitas beras menjadi 3 label yaitu premium, medium, dan tidak layak. Dari permasalahan tersebut, kami memiliki solusi yaitu membuat model *machine learning* menggunakan algoritma *Convolutional Neural Network* (CNN) dengan *library* TensorFlow dan Keras, serta arsitektur MobileNet untuk membuat model dengan akurasi yang baik. Kemudian dibuat model *deployment* untuk *on-device* dengan format tflite. Alur kerja penelitian dapat dilihat pada Gambar 1.



Gambar 1. Flowchart Pengembangan Model *Machine Learning*

Proses pengembangan dan konversi model *Machine Learning* dilakukan dengan mengikuti langkah-langkah dalam *block diagram* tersebut. Pertama, dataset yang akan digunakan diunggah ke Google Drive. Setelah itu, dataset diimpor dari Google Drive ke lingkungan kerja Google Colab untuk memudahkan proses pemrosesan dan pembangunan model. Setelah model selesai dibangun dan diuji di Google Colab, file ipynb yang berisi kode dan hasil pekerjaan disimpan kembali ke Google Drive. Selanjutnya, pengguna membuat dan menulis kode program menggunakan bahasa Python di Google Colab, dan mengimpor *framework* TensorFlow untuk membangun model *Machine Learning*. Dalam proses pembangunan model, arsitektur model dibangun menggunakan MobileNet, yang merupakan arsitektur jaringan saraf yang efisien dan sering digunakan untuk aplikasi *computer vision* pada perangkat seluler. Setelah model selesai dibangun dan dilatih menggunakan data, model tersebut dikonversi dari format yang digunakan di Google Colab ke format tflite. Format tflite adalah format yang dioptimalkan untuk dijalankan pada perangkat seluler, sehingga model dapat diimplementasikan dengan efisien dalam aplikasi Android. Dengan demikian, proses pengembangan dan konversi model *Machine Learning* ini memungkinkan model untuk diimplementasikan dan dijalankan dengan baik pada perangkat seluler dalam format yang sesuai.

2.1.1. Pengolahan Data

Dataset yang kami gunakan terdiri dari 3 label yaitu premium, medium, dan tidak layak. Tahap pertama dalam proses pengolahan data adalah data mentah yang berisi 1800 gambar. Data mentah ini kemudian melewati proses "Data Cleaning" yang bertujuan untuk membersihkan dan mempersiapkan data agar sesuai dengan kebutuhan analisis lebih lanjut. Proses *data cleaning* meliputi pemrosesan untuk menghilangkan data yang hilang, menangani data yang duplikat, atau melakukan konversi format data.

Setelah melalui proses *data cleaning*, data 1800 gambar tersebut didistribusikan untuk keperluan *training* dan validasi model *machine learning*.

a. *Training Data:*

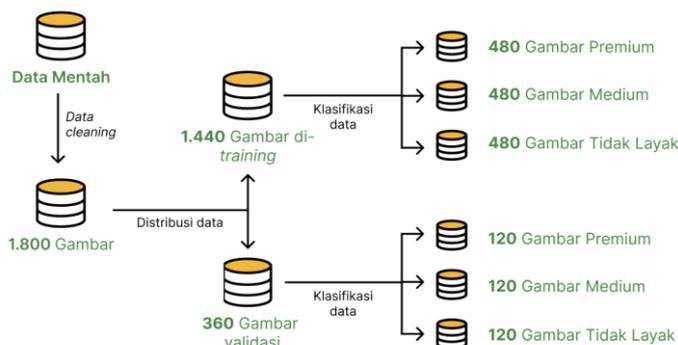
Dari 1800 gambar, sebanyak 1440 gambar digunakan sebagai data *training* untuk melatih model *machine learning*. Data *training* ini diklasifikasikan ke dalam tiga kategori, yaitu:

- 480 gambar kategori "premium"
- 480 gambar kategori "medium"
- 480 gambar kategori "tidak layak"

b. *Validasi Data:*

Selanjutnya, sebanyak 360 gambar digunakan sebagai data validasi untuk menguji peforma model *machine learning* yang telah dilatih. Data validasi ini juga diklasifikasikan ke dalam tiga kategori, yaitu:

- 120 gambar kategori "premium"
- 120 gambar kategori "medium"
- 120 gambar kategori "tidak layak"



Gambar 2. Proses Pengolahan Data

Berdasarkan langkah-langkah yang ditunjukkan Gambar 2, dataset telah dibagi menjadi data *training* dan data validasi dengan distribusi yang telah ditentukan untuk melatih dan menguji model *machine learning*. Data yang telah melalui proses pengolahan ini siap digunakan untuk melatih model dan melakukan evaluasi dalam tugas klasifikasi berdasarkan kategori yang telah ditentukan.

2.1.2. Pelatihan Model

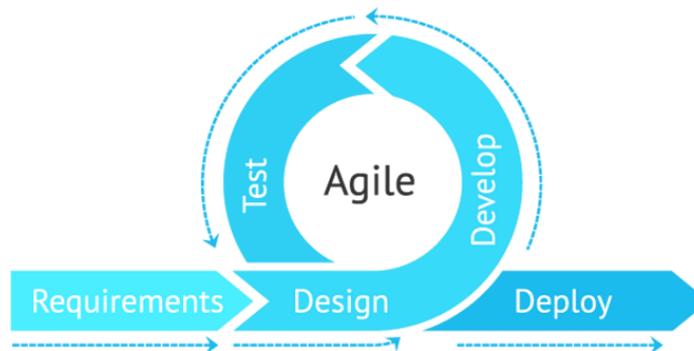
Model dilatih menggunakan arsitektur MobileNet dengan total 3.523.907 parameter, yang mana 295,043 parameter dapat dilatih (*trainable*) dan 3.228.864 parameter tidak dapat dilatih (*non-trainable*). Arsitektur MobileNet digunakan sebagai dasar model dalam proses *transfer learning*, dimana sebagian besar parameter telah di-*pre-trained* pada tugas penglihatan komputer umum. Dengan demikian, menggunakan model MobileNet sebagai model dasar memungkinkan proses pelatihan lebih efisien dan mengurangi kebutuhan akan dataset pelatihan yang besar untuk mencapai hasil yang baik. Parameter *trainable* yang relatif kecil menandakan proses pelatihan lebih fokus pada penyesuaian model dengan data spesifik dari tugas yang dihadapi. Hasil dari proses pelatihan ini akan menghasilkan model pendeteksi kualitas beras yang dapat digunakan untuk mengklasifikasikan gambar beras ke dalam kategori premium, medium, dan tidak layak berdasarkan fitur-fitur visual yang diambil dari dataset pelatihan. Dengan menggunakan arsitektur MobileNet yang telah di-*pre-trained*, model dapat mencapai tingkat akurasi yang tinggi dan kemampuan generalisasi yang baik pada tugas pendeteksian kualitas beras.

Setelah melalui tahap pengujian dan model dipastikan memiliki tingkat kepercayaan tinggi, maka selanjutnya model dapat dikonversi menggunakan TF Lite Converter menjadi file *flatbuffer* yang bisa digunakan pada perangkat seluler dan perangkat *embedded* lainnya. File model TF Lite ini memiliki ekstensi ".tflite" dan

dapat diimplementasikan pada perangkat yang menjalankan sistem operasi Android, iOS, atau *platform embedded* lainnya [9]. Konversi ini akan mengubah model yang sebelumnya berformat TensorFlow (biasanya dengan ekstensi .pb) menjadi format yang dapat dijalankan menggunakan TensorFlow Lite.

2.2. Model Pengembangan Aplikasi

Pada penelitian ini, aplikasi dikembangkan menggunakan pendekatan metode Agile seperti yang ditunjukkan pada Gambar 3.



Gambar 3. Tahapan Metode Agile

Agile merupakan salah satu metode yang efektif dalam proses pengembangan perangkat lunak [5]. Pendekatan Agile untuk pengembangan aplikasi *mobile* menggambarkan pendekatan iteratif dan inkremental yang melibatkan tim yang otonom dan kolaboratif dalam proses pembangunan perangkat lunak [6]. Metode ini memiliki beberapa tahapan, yaitu *requirements*, *design*, *develop*, *test*, dan *deploy*.

1. *Requirements* (Kebutuhan)

Requirements adalah tahapan melakukan penilaian terhadap pengguna untuk mendapatkan pemahaman rinci mengenai kebutuhan perangkat lunak dari pengguna [7]. Tahap ini berkaitan dengan kebutuhan-kebutuhan dari perangkat lunak maupun keras pengguna. Pada tahap ini menghasilkan data analisis yaitu spesifikasi perangkat lunak, fitur-fitur aplikasi, dan lain-lain.

2. *Design* (Perancangan)

Setelah kebutuhan dikumpulkan, pengembang melakukan perancangan perangkat lunak berdasarkan requirement yang telah didefinisikan. Pada tahap ini dilakukan rancangan sistem yang mencakup berbagai aspek, seperti desain arsitektur sistem, desain proses bisnis, desain database, dan desain antarmuka pengguna [7]. Perancangan dapat mencakup desain antarmuka pengguna, arsitektur sistem, dan rancangan komponen perangkat lunak lainnya. Spesifikasi yang dibuat cukup terperinci dengan menggunakan UML (*Unified Modelling Language*), sehingga pada tahap implementasi langsung mengikuti apa yang telah ditentukan dalam tahap perancangan. UML adalah bahasa pemodelan visual yang digunakan untuk menentukan, menggambarkan, membangun, dan mendokumentasikan perangkat lunak.

3. *Develop* (Pengembangan)

Pada tahap ini, di mana perancangan aplikasi diimplementasikan melalui proses pembuatan kode dan database [7]. Pengembangan dilakukan dalam iterasi yang singkat dan dapat mencakup beberapa kebutuhan dalam setiap iterasinya. Pengimplementasian perancangan aplikasi ini menggunakan bahasan pemrograman Kotlin dan teknologi *machine learning* berbasis metode *convolutional neural network* (CNN) dengan arsitektur MobileNet.

4. *Test* (Pengujian)

Tahap ini dilakukan dengan tujuan untuk mengurangi kesalahan (*error*) dan memastikan bahwa hasil keluaran telah sesuai dengan kebutuhan pengguna [7]. Pengujian yang dilakukan dalam perancangan aplikasi ini adalah pengujian *black box*. Pengujian *black box* adalah metode pengujian fungsionalitas sebuah aplikasi di mana penguji hanya mengevaluasi *input* dan *output* aplikasi tanpa meninjau detail internal dari sistem tersebut.

5. *Deploy*

Setelah fitur-fitur telah selesai diuji dan dinyatakan siap untuk produksi, dilakukan penerapan atau penyebaran perangkat lunak kepada pengguna akhir atau lingkungan produksi. Setelah fitur-fitur selesai diuji dan dinyatakan siap untuk digunakan oleh pengguna.

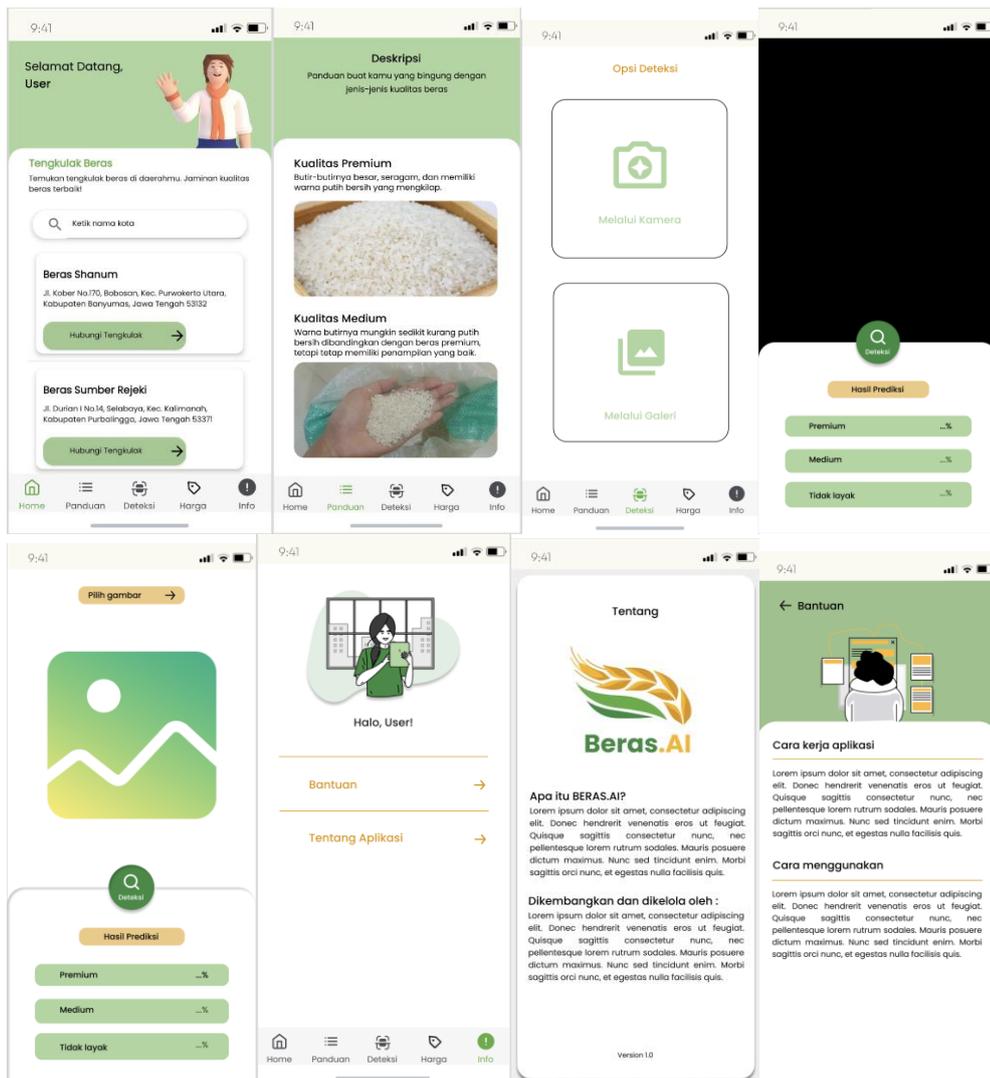
3. HASIL DAN PEMBAHASAN

Penelitian ini menghasilkan aplikasi pendeteksi android pendeteksi kualitas beras. Aplikasi ini memberikan kemudahan dalam proses pengecekan deteksi kualitas beras.

3.1. Hasil Pengembangan Aplikasi Android

Penelitian ini menghasilkan Aplikasi Android Pendeteksi Kualitas Beras berbasis Machine Learning yang dibangun dengan bahasa pemrograman Kotlin dan Android Studio sebagai IDE. Gambar 4 menunjukkan hasil desain UI/UX aplikasi android pendeteksi kualitas beras. Desain UI/UX bertujuan untuk menciptakan antarmuka aplikasi yang menarik, intuitif, dan mudah digunakan oleh pengguna. Dalam proses ini, elemen-elemen visual seperti tata letak, warna, ikon, dan interaksi antarmuka dipercanggih dengan cermat untuk meningkatkan pengalaman pengguna secara keseluruhan.

Berdasarkan Gambar 4, aplikasi ini memiliki beberapa halaman utama, yaitu Halaman *Home* sebagai halaman awal (gambar paling kiri ke-1), Halaman Panduan yang memberikan informasi tentang karakteristik kualitas beras (gambar kedua dari kiri), dan Halaman Deteksi (gambar ketiga dari kiri) yang memungkinkan pengguna untuk mendeteksi kualitas beras melalui kamera atau galeri. Kemudian ada halaman harga yang menampilkan informasi harga beras di setiap provinsi, sementara Halaman Info berisi tombol yang mengarahkan ke Halaman Bantuan dan Halaman Tentang Aplikasi. Halaman Bantuan memberikan informasi tentang cara kerja aplikasi dan cara penggunaan dalam mendeteksi beras. Halaman Tentang Aplikasi memberikan informasi tentang teknologi yang digunakan dalam pembuatan aplikasi dan informasi developer.



Gambar 4. Desain UI/UX Aplikasi

3.1.1. Pengujian Blackbox

Setelah Aplikasi Android Pendeteksi Kualitas Beras Berbasis *Machine Learning* selesai dibangun, maka tahap selanjutnya adalah melakukan pengujian. Metode pengujian yang kami gunakan adalah *Blackbox testing*. Metode Blackbox testing merupakan salah satu metode yang mudah digunakan karena hanya memerlukan batas bawah dan batas atas dari data yang diharapkan. Estimasi banyaknya data uji dapat dihitung melalui banyaknya field data entri yang akan diuji, aturan entri yang harus dipenuhi serta kasus batas atas dan batas bawah yang memenuhi. Dan dengan metode ini dapat diketahui jika fungsionalitas masih dapat menerima masukan data yang tidak diharapkan maka menyebabkan data yang disimpan kurang valid [10]. Hasil dari pengujian *blackbox* dapat dilihat pada data dibawah ini.

1.) Pengujian terhadap Halaman *Home*

Tabel 1. Hasil Pengujian *Blackbox Testing* Halaman *Home*

| No | Skenario Pengujian | Test Case | Hasil yang diharapkan | Hasil Uji |
|----|--|-----------|--|-----------|
| 1. | User mengakses halaman home | - | Aplikasi menampilkan halaman home yang berisi informasi mengenai tengkulak beras | Valid |
| 2. | User mencari informasi tengkulak berdasarkan nama kota menggunakan fitur search. | Surabaya | Aplikasi menampilkan daftar tengkulak yang berada di kota Surabaya | Valid |
| 3. | User menekan tombol “Hubungi Tengkulak” | - | Aplikasi kemudian beralih ke room chat WhatsApp. | Valid |

Tabel 1 menunjukkan hasil pengujian halaman home yang dibagi menjadi 3 skenario pengujian yaitu (1) akses tombol home; (2) pencarian informasi tengkulak; dan (3) tombol hubungi tengkulak. Semua hasil uji menunjukkan valid untuk tiga skenario tersebut. Tabel 2 menunjukkan hasil pengujian halaman deteksi yang dibagi menjadi 3 skenario yaitu (1) menekan halaman deteksi; (2) menekan tombol kamera deteksi; dan (3) menekan tombol deteksi dari galeri. Semua hasil uji menunjukkan valid untuk tiga skenario tersebut.

2.) Pengujian terhadap Halaman Deteksi

Tabel 2. Hasil Pengujian *Blackbox Testing* Halaman Deteksi

| No | Skenario Pengujian | Test Case | Hasil yang diharapkan | Hasil Uji |
|----|--|-----------|--|-----------|
| 1. | User mengakses halaman Deteksi | - | Aplikasi menampilkan dua tombol pilihan untuk mengakses fitur deteksi melalui kamera dan galeri. | Valid |
| 2. | User menekan tombol Deteksi Melalui Kamera | - | Aplikasi beralih ke menu deteksi melalui kamera. | Valid |
| 3. | User menekan tombol Deteksi Melalui Galeri | - | Aplikasi beralih ke menu deteksi melalui galeri | Valid |

Tabel 3 menunjukkan hasil pengujian model deteksi melalui kamera yang dibagi menjadi 3 skenario pengujian yaitu (1) deteksi beras kualitas premium; (2) deteksi beras kualitas medium; dan (3) deteksi beras kualitas tidak layak. Semua hasil uji menunjukkan valid untuk tiga skenario tersebut. Pengujian black box terakhir ditunjukkan Tabel 4 menunjukkan hasil pengujian model deteksi melalui galeri yang dibagi menjadi 3 skenario yaitu (1) mengunggah gambar beras kualitas premium; (2) mengunggah gambar beras kualitas medium; dan (3) mengunggah gambar beras kualitas tidak layak. Semua hasil uji menunjukkan valid untuk tiga skenario tersebut.

3.) Pengujian terhadap Halaman Deteksi Melalui Kamera

Tabel 3. Hasil Pengujian *Blackbox Testing* Halaman Deteksi Melalui Kamera

| No | Skenario Pengujian | Test Case | Hasil yang diharapkan | Hasil Uji |
|----|--|---|--|-----------|
| 1. | Mengarahkan kamera ke objek beras berkualitas premium. | Beras berkualitas premium, yang memiliki karakteristik butir bersih, putih, dan memiliki derajat sosoh yang baik. | Menampilkan hasil deteksi premium dengan akurasi tertinggi | Valid |
| 2. | Mengarahkan kamera ke objek beras berkualitas medium. | Beras berkualitas medium yang memiliki karakteristik butir sedikit kusam, tidak terlalu putih, dan derajat sosoh yang | Menampilkan hasil deteksi medium dengan akurasi tertinggi | Valid |

| | | | | |
|----|--|--|--|-------|
| 3. | Mengarahkan kamera ke objek beras berkualitas tidak layak. | cukup baik. Beras berkualitas tidak layak yang memiliki karakteristik butir kusam, terdapat benda asing (kutu dan gabah), derajat sosoh tidak baik. | Menampilkan hasil deteksi tidak layak dengan akurasi tertinggi | Valid |
|----|--|--|--|-------|

4.) Pengujian terhadap Halaman Deteksi Melalui Galeri

Tabel 4. Hasil Pengujian *Blackbox Testing* Halaman Deteksi Melalui Galeri

| No | Skenario Pengujian | Test Case | Hasil yang diharapkan | Hasil Uji |
|----|---|---|---------------------------------------|-----------|
| 1. | Mengunggah gambar beras berkualitas premium | Gambar beras berkualitas premium, yang memiliki karakteristik butir bersih, putih, dan memiliki derajat sosoh yang baik. | Menampilkan hasil deteksi premium | Valid |
| 2. | Mengunggah gambar beras berkualitas medium | Gambar beras berkualitas medium yang memiliki karakteristik butir sedikit kusam, tidak terlalu putih, dan derajat sosoh yang cukup baik. | Menampilkan hasil deteksi medium | Valid |
| 3. | Mengunggah gambar beras berkualitas tidak layak | Gambar beras berkualitas tidak layak yang memiliki karakteristik butir kusam, terdapat benda asing (kutu dan gabah), derajat sosoh tidak baik | Menampilkan hasil deteksi tidak layak | Valid |

Hasil pengujian *Blackbox* menunjukkan bahwa semua menu yang telah dibuat memberikan hasil yang valid sesuai dengan yang diharapkan oleh pengembang aplikasi.

3.1.2. *Deployment*

Proses pelatihan model dilakukan selama 25 epoch. Epoch mengindikasikan satu siklus di mana algoritma *machine learning* ‘mempelajari’ dataset *training* yang akan diproses [8]. Satu epoch berarti seluruh dataset pelatihan telah dilewatkan satu kali melalui model saat proses pelatihan. Ketika proses pelatihan dimulai, model akan memproses data pelatihan berulang kali selama beberapa epoch untuk mengoptimalkan bobot dan biasanya.

3.2. Hasil Pengujian Model Deteksi

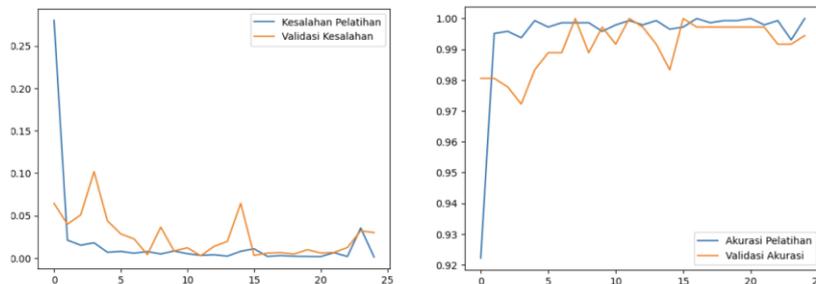
Proses pelatihan model dilakukan selama 25 epoch. Epoch mengindikasikan satu siklus di mana algoritma *machine learning* ‘mempelajari’ dataset *training* yang akan diproses [8]. Satu epoch berarti seluruh dataset pelatihan telah dilewatkan satu kali melalui model saat proses pelatihan. Ketika proses pelatihan dimulai, model akan memproses data pelatihan berulang kali selama beberapa epoch untuk mengoptimalkan bobot dan biasanya.

3.2.1. Hasil Data Latih

Pada epoch ke-25, hasil evaluasi model pada data latih sebagai berikut:

1. *Loss* pada data pelatihan (*training loss*) adalah 0.0012.
2. *Loss* merupakan ukuran seberapa baik model memprediksi label yang benar dari data pelatihan. Semakin rendah nilai *loss*, semakin baik model mempelajari pola-pola dari data dan semakin akurat model dalam memprediksi label.
3. Akurasi pada data pelatihan (*training accuracy*) adalah 1.000 (atau 100%).
4. Akurasi merupakan presentase dari data pelatihan yang diprediksi dengan benar oleh model. Akurasi yang mencapai 100% menunjukkan bahwa pada akhir epoch ke-25, model dapat memprediksi seluruh data pelatihan dengan benar.
5. *Loss* pada data validasi (*validation loss*) adalah 0.0299.
6. *Validation loss* adalah ukuran seberapa baik model dapat menggeneralisasi pada data yang belum pernah dilihat sebelumnya (data validasi). Semakin rendah nilai *validation loss*, semakin baik model dalam menghindari *overfitting* dan mampu memprediksi data baru dengan akurat.
7. Akurasi pada data validasi (*validation accuracy*) adalah 0.9944 (atau 99.44%).
8. *Validation accuracy* merupakan presentase dari data validasi yang diprediksi dengan benar oleh model. Akurasi yang tinggi pada data validasi menunjukkan bahwa model telah berhasil menggeneralisasi dengan baik dan dapat memprediksi data baru dengan akurat.

Hasil dari epoch ke-25 menunjukkan bahwa model telah mengalami proses pelatihan yang baik dan berhasil mencapai akurasi yang tinggi pada data pelatihan dan data validasi. Hasil kesalahan dan akurasi pada masing-masing epoch dapat ditunjukkan pada Gambar 5.



Gambar 5. Grafik Kesalahan dan Akurasi Model.

3.2.2. Hasil Deteksi

Setelah model terbentuk melalui proses pelatihan, selanjutnya dilakukan pengujian untuk mengevaluasi kemampuan deteksi kualitas beras dengan akurat. Hasil *testing* menunjukkan bahwa model memberikan prediksi dengan tingkat kepercayaan yang tinggi terhadap kelas tertentu. Berdasarkan *output* dari model, diperoleh probabilitas prediksi sebagai berikut:

- Probabilitas kelas “Premium”: 8.4767744e-06
- Probabilitas kelas “Medium”: 2.5625873e-09
- Probabilitas kelas “Tidak layak”: 9.9999154e-01

Dari probabilitas tersebut, dapat diinterpretasikan bahwa model memberikan prediksi paling tinggi pada kelas “Tidak Layak” dengan probabilitas sekitar 0.9999 atau 99.99%. Sementara itu, probabilitas prediksi untuk kelas “Premium” dan “Medium” sangat rendah, mendekati nol. Waktu yang dibutuhkan untuk melakukan prediksi satu gambar adalah 44ms/step menunjukkan kecepatan dan efisiensi model dalam melakukan inferensi atau pengujian. Waktu yang singkat ini menandakan bahwa model dapat bekerja dengan cepat dan dapat diimplementasikan secara *real-time* pada aplikasi seluler atau platform lainnya. Berdasarkan hasil testing ini, dapat disimpulkan bahwa model telah berhasil mengidentifikasi gambar sample sebagai kelas “Tidak layak” dengan tingkat kepercayaan yang tinggi. Hasil ini menunjukkan bahwa model yang telah dilatih dan diuji memiliki performa yang baik dalam tugas pengidentifikasian kualitas beras.

4. KESIMPULAN

Penelitian ini menghasilkan aplikasi pendeteksi kualitas beras berbasis machine learning dengan menggunakan metode *Convolutional Neural Network* (CNN) dan arsitektur MobileNet. Aplikasi ini dibangun menggunakan bahasa pemrograman Kotlin dan IDE Android Studio, dengan penerapan pendekatan Agile dalam proses pengembangannya. Hasil uji Blackbox menunjukkan bahwa aplikasi ini berhasil melewati pengujian dengan hasil yang valid sesuai harapan pengembang.

Proses pelatihan model menggunakan metode CNN pada arsitektur MobileNet mencapai tingkat akurasi yang memuaskan dalam identifikasi kualitas beras. Aplikasi ini mampu mengenali dan mengklasifikasikan objek beras dengan baik, memberikan prediksi yang akurat, dan berpotensi memberikan kontribusi positif dalam mendukung industri pertanian serta meningkatkan kualitas pangan.

Saran untuk penelitian selanjutnya adalah mengembangkan hasil prediksi dengan penambahan informasi yang lebih rinci dan relevan tentang kualitas beras yang dideteksi. Pengembangan informasi ini akan memberikan manfaat lebih banyak bagi pengguna dalam memahami karakteristik beras yang terdeteksi dan mendukung keputusan terkait pemilihan beras yang sesuai dengan kebutuhan dan preferensi.

DAFTAR PUSTAKA

- [1] W. M. Handani, N. Kusnadi and D. Rachmina, "Prospek Swasembada Beras di Provinsi Kalimantan Timur," *Jurnal Agribisnis Indonesia*, vol. 9, no. 1, pp. 67-68, 2021.
- [2] J. David and T. Kartinaty, "Karakteristik Mutu Beras di Berbagai Penggilingan Pada Sentra Padi di Kalimantan Barat," *Journal TABARO Agriculture Science*, vol. 3, no. 1, pp. 276-286, 2019.
- [3] D. Lestari, N. Fadillah and A. Ihsan, "Sistem Deteksi Kualitas Beras Berdasarkan Warna Menggunakan

-
- Fuzzy C-Means Clustering* Guna Membantu Tingkat Pengetahuan Masyarakat," *Jurnal Nasional Informatika dan Teknologi Jaringan*, vol. 3, no. 2, pp. 137-142, 2019.
- [4] Fadhillah, A. Kuswandi and P. Haryono, "Peranan Aplikasi Android Dalam Peningkatan Kualitas Pelayanan Sekolah di Pesantren Persis Kota Tasikmalaya," *Jurnal Manajemen Pendidikan*, vol. 8, no. 1, pp. 22-32, 2021.
- [5] R. F. Malik, F. Muhammad and P. Rahmanto, "Sistem Informasi Manajemen Laboratorium Menggunakan Metode *Agile* dengan Konsep *Model-View-Controller Data Access Object*," Universitas Sriwijaya, Palembang.
- [6] A. N. Yusril, I. Larasati and P. A. Zukri, "Systematic Literature Review Analisis Metode *Agile* dalam Pengembangan Aplikasi *Mobile*," *Jurnal Sistem Informasi*, vol. 10, no. 2, pp. 369-380, 2021.
- [7] I. Mahendra and D. T. E. Yanto, "*Agile Development Methods* Dalam Pengembangan Sistem Informasi Pengajuan Kredit Berbasis Web (Studi Kasus : Bank BRI Unit Kolonel Sugiono)," *Jurnal Teknologi dan Open Source*, vol. 1, no. 2, pp. 13-24, 2018.
- [8] M. Wasil, Harianto and Fathurrahman, "Pengaruh *Epoch* pada Akurasi menggunakan *Convolutional Neural Network* untuk Klasifikasi fashion dan Furniture," *Jurnal Informatika dan Teknologi*, vol. 5, no. 1, pp. 53-61, 2022.
- [9] Roslidar, M. R. Syahputra, R. Muharar and F. Arnia, "Adaptasi Model CNN Terlatih pada Aplikasi Bergerak untuk Klasifikasi Citra Ternal Payudara," *Jurnal Rekayasa Elektrika*, vol. 18, no. 3, pp. 185-192, 2022.
- [10] W. N. Cholifah, Yulianingsih and S. M. Sagita, "PENGUJIAN *BLACK BOX TESTING* PADA APLIKASI ACTION & STRATEGY BERBASIS ANDROID DENGAN TEKNOLOGI PHONEGAP," *Jurnal String*, vol. 3, no. 2, pp. 206-210, 2018.