

Implementation of CNN Trained Model on Coffee Type Detection Android App with Photo Input of Beans, Fruits and Leaves

M. Taufik Hidayat¹, Pradita Eko Prasetyo Utomo², Benedika Ferdian Hutabarat³

^{1,2,3}Program Studi Sistem Informasi Fakultas Sains Dan Teknologi Universitas Jambi

^{1,2,3}Jl. Jambi – Muara Bulian No.KM. 15 Mendalo Darat, Jambi Luar Kota, Jambi

e-mail: th309015@gmail.com, pradita.eko@unja.ac.id, benedika@unja.ac.id

Abstract

Coffee is the most consumed type of drink in the world. Each type of coffee has different physical characteristics from leaves, fruits to seeds. Now technology is needed in the world of agriculture in making decisions. To determine the type of coffee with fission characteristics, there are still many people who do not understand in distinguishing the physical characteristics of coffee plants. In this case, an application was developed using the RAD method by utilizing the flutter framework and the Convolutional Neural Network model that has been trained. The pre-train model used is NasNet Mobile with a dataset of 900 photos and 100 epochs with early-stopping utilization and heti at epoch 55 with an accuracy of 90.67%. In this study, implementing existing models into Android applications using the Flutter framework. With the implementation process carried out by the application can help the detection process using an android device. The implementation results get good test results with a score of 0.97. This application can help the process of identifying the type of coffee and minimize errors in identifying directly.

Keywords: CNN, Flutter, Coffee, Android, Implementation

Implementasi Model Terlatih CNN pada Aplikasi Android Pendeteksi Jenis Kopi dengan Input Foto Biji, Buah dan Daun

Abstrak

Kopi merupakan jenis minuman yang paling banyak dikonsumsi di dunia. Setiap jenis kopi memiliki karakter fisik yang berbeda-beda dari daun, buah hingga biji. Kini teknologi sangat dibutuhkan dalam dunia pertanian dalam mengambil keputusan. Untuk menentukan jenis kopi dengan karakter fisi, masih banyak orang belum memahami dalam membedakan karakteristik fisik tanaman kopi. Dalam hal ini dikembangkan aplikasi menggunakan metode RAD dengan memanfaatkan framework flutter dan model Convolutional Neural Network yang telah dilatih.

<https://doi.org/10.31849/digitalzone.v15i1.19563>

Digital Zone is licensed under a Creative Commons Attribution International (CC BY-SA 4.0)

Model *pre-train* yang digunakan adalah *NasNet Mobile* dengan dataset 900 foto dan 100 epoch dengan pemanfaatan *early-stopping* dan berhenti pada epoch ke 55 dengan akurasi 90,67%. Dalam penelitian ini melakukan implementasi model yang telah ada kedalam aplikasi android dengan menggunakan framework flutter. Dengan proses implementasi yang dilakukan aplikasi dapat membantu proses deteksi menggunakan perangkat android. Hasil implementasi mendapatkan hasil pengujian yang baik dengan skor 0.97. Aplikasi ini dapat membantu proses identifikasi jenis kopi dan meminimalisir kesalahan dalam melakukan identifikasi secara langsung.

Kata kunci: CNN, Flutter, Kopi, Android, Implementasi

1. Pendahuluan

Di Indonesia yang merupakan iklim tropis dengan produksi hasil pertanian tertinggi di dunia menempati urutan ke empat produksi kopi terbanyak di dunia dengan total produksi mencapai 711,3 ton per tahun dengan brazil menjadi urutan pertama negara produksi kopi terbanyak di dunia dengan total peroduksi 3103,5 ton per tahun pada tahun 2015-2020 [1]. dalam hal ini Indonesia memiliki potensi pertanian kopi yang sangat besar bagi dunia. Petani memerlukan kemampuan dalam menentukan jenis kopi yang akan dijadikan sebagai komoditas pertanian. Sehingga diperlukan alat yang dapat mempermudah dalam membedakan jenis kopi dengan ciri visual dari tanaman kopi.

Dalam melakukan identifikasi jenis kopi, belum terdapat alat yang bisa membantu dalam bidang pertanian untuk melakukan identifikasi jenis kopi. Hal ini membutuhkan inovasi untuk membantu bidang pertanian dalam melakukan identifikasi jenis kopi. Dengan memanfaatkan teknologi seperti perangkat mobile yang dapat digunakan secara berpindah-pindah dapat mempermudah dalam sektor pertanian. Pemanfaatan perangkat android dan *Machine Learning* menjadi pilihan dalam membangun inovasi dalam bidang pertanian salah satunya pada pertanian kopi untuk melakukan identifikasi jenis tanaman kopi. CNN merupakan algoritma *deep learning* dari *Artificial Neural Network* (ANN) dengan meniru jaringan syaraf pada otak untuk melakukan klasifikasi gambar [2].

Android menjadi sistem operasi paling banyak digunakan dengan total peredaran dunia mencapai 72,95% dan iOS 26,27% [3]. Implementasi model terlatih CNN (*Convolutional Neural Network*) pada aplikasi android sangat diperlukan dalam membantu membandingkan jenis kopi dimanapun dan kapapun oleh siapapun. Model yang akan diimplementasikan memiliki format Tensorflowlite (.tflite) untuk mengurangi ukuran model dengan menjalankan model aslinya berupa model Tensorflow yang telah dilatih sebelumnya dan sebagai alat yang digunakan dalam menyimpan model Tensorflow yang kemudian dapat diimplementasikan pada aplikasi mobile yang dikembangkan [4]. Dengan format model yang telah diubah menjadi Tensorflowlite maka akan dapat mengurangi ukuran aplikasi yang cukup signifikan dengan pemrosesan gambar yang cepat tanpa mengurangi akurasi dari model yang telah dilatih.

NasNet Mobile merupakan salah satu model *pre-train* dengan arsitektur CNN yang memiliki akurasi pelatihan yang sangat baik dan dirancang untuk dipergunakan dalam perangkat mobile karena ukuran model yang dihasilkan relative kecil sehingga ringan untuk digunakan dalam perangkat mobile. Model yang diperoleh sebelumnya telah dilatih menggunakan dataset biji, buah dan daun kopi dengan total dataset 900 foto dengan terdiri dari tiga kelas (Arabika, Robusta dan Liberika) dan tiga parameter (biji, buah dan daun) untuk setiap kelas.

Dalam melakukan implementasi model pada aplikasi android akan menggunakan metode RAD (*Rapid Application Development*). Metode RAD merupakan metode pengembangan perangkat lunak dengan sistem kerja yang singkat namun mengutamakan kenyamanan pengguna dengan melibatkan pengguna dalam merancang sistem sehingga dapat menghasilkan rancangan sesuai dengan keinginan pengguna akhir. Implementasi dilakukan dengan

menggunakan *framework* Flutter dengan Bahasa pemrograman Dart. Metode ini merujuk pada penelitian terdahulu oleh Arianto pada penelitian “Perancangan Aplikasi Pengenalan dan Pembelajaran Pertanian Milenial Hidroponik Berbasis Android Flutter & Dart dengan Metode RAD” [5]. Dengan menggunakan metode RAD akan dapat menghasilkan aplikasi dengan kualitas yang baik dan proses pengerjaan yang lebih singkat.

Aplikasi deteksi jenis kopi dapat meningkatkan akurasi dalam melakukan identifikasi jenis kopi dengan menggunakan citra visual yang sebelumnya seorang petani, konsumen maupun produseren produk olahan kopi membutuhkan kemandirian untuk melakukan identifikasi jenis kopi, dengan ini dapat membantu proses identifikasi supaya mengurangi Tingkat kesalahan dalam identifikasi jenis kopi. Selain itu juga dapat mengurangi terjadinya miskomunikasi antara penjual bibit, biji, maupun buah kopi yang memungkinkan terjadinya penipuan terkait jenis kopi tertentu.

2. Metode Penelitian

Metode yang digunakan dalam penelitian ini menggunakan metode pengembangan sistem RAD (*Rapid Application Development*) untuk melakukan pengembangan aplikasi android deteksi jenis kopi.

2.1. Rapid Application Development (RAD)

Rapid Application Development merupakan salah satu metode dari *Software Development Life Cycle* (SDLC) dalam tahapan pengembangan perangkat lunak yang memiliki tahapan pengembangan yang singkat [6]. Dalam pengembangan menggunakan metode RAD memiliki tiga tahapan utama yaitu [7]:



Gambar 1. Tahapan Penelitian

2.2.1. Perencanaan kebutuhan (*Requirement planning*)

Tahapan ini penulis melakukan analisis terhadap tujuan pengembangan aplikasi serta identifikasi terhadap kebutuhan informasi untuk mencapai tujuan. dalam perencanaan kebutuhan dilakukan dengan melakukan wawancara kepada petani kopi liberika desa Mekar Jaya, Owner caffe Motokoffie dan widyaiswara Balai Pelatihan Pertanian Jambi. Dari hasil wawancara akan menghasilkan kebutuhan fungsional sistem yang kemudian digunakan dalam perancangan sistem.

2.2.2. Pemodelan (*Workshop design*)

Tahapan ini dilakukan desain rancangan dan perbaikan apabila adanya ketidak sesuaian antara rancangan yang dibuat dengan keinginan pengguna dengan melibatkan pengguna untuk persetujuan pengembangan lebih lanjut. Dalam proses pemodelan dilakukan desain terhadap *usecase* dan *Activity diagram* untuk selanjutnya digunakan dalam tahapan implementasi. *Usecase* berisikan *case* yang dapat dilakukan oleh pengguna pada aplikasi. Sedangkan *activity diagram* adalah tahapan dari *case* yang dilakukan oleh pengguna pada aplikasi.

2.2.3. Penerapan (*Implementation*)

Setelah desain sistem disetujui oleh pengguna yang kemudian akan dilanjutkan oleh pengembang melakukan pemilihan model deteksi yang ingin digunakan dan kemudian pengkodean sesuai dengan hasil rancangan yang telah disetujui oleh pengguna. Tensorflow merupakan salah satu *framework deep learning* yang digunakan dalam mengembangkan berbagai program *Artificial Intelligence* (AI). Salah satu program AI yang dikembangkan menggunakan *framework tensorflow* adalah klasifikasi citra yang digunakan

dalam memprediksi kelas suatu citra visual dari gambar[8]. Tensorflow lite merupakan library dari machine learning yang dirancang untuk mengenali suatu objek [9]. Tensorflow lite merupakan model hasil convert model tensorflow untuk mendapatkan model yang lebih ringan, efisien dan dapat dijalankan dalam perangkat dengan penyimpanan yang tidak terlalu besar [10]. Hal ini dapat meningkatkan efisiensi dalam penerapan model deteksi pada perangkat android.

Pengkodean menggunakan framework flutter dan bahasa pemrograman dart. Flutter merupakan *Software Development Kit* (SDK) yang dikembangkan Google untuk memudahkan dalam pengembangan aplikasi android dan iOS tanpa harus mempelajari Bahasa untuk pengembangan dua sistem operasi yang berbeda[11]. Penggunaan flutter dalam pengembangan aplikasi dapat memperoleh aplikasi dengan antarmuka yang menarik, responsive, dan dapat berjalan pada multiplatform [12]. Dart merupakan Bahasa pemrograman yang dirancang oleh Lars Bak dan Kasper Lund dan dikembangkan oleh Google[11]. Flutter memiliki *package* yang dapat memudahkan proses pengembangan aplikasi yaitu flutter *package* dan flutter *package dev* [13]. Dalam proses implementasi memerlukan *package* flutter berupa *tflite_flutter* dan *tflite_flutter_helper* untuk memanggil model dan interpretasi model yang telah di-*import* kedalam project flutter [14]. *Package* tersebut kemudian dipanggil dalam kode program, selanjutnya dilakukan pemanggilan model dalam format *tflite* dan label kelas dalam bentuk *txt*. Apabila model telah berhasil di panggil dalam kode program kemudian melakukan pengambilan gambar menggunakan *image_picker* dan memproses gambar pada model yang tanam dalam program sehingga dapat menghasilkan prediksi berupa persentase *confidence* untuk tiap kelas berupa jenis kopi arabika, robusta dan liberika.

2.2. Pengujian

Pengujian perangkat lunak merupakan tahapan pengujian program yang memiliki tujuan dalam mencari kesalahan atau error dari sistem yang dikembangkan [15]. Terdapat tiga pengujian perangkat lunak yaitu *White-box testing*, *Black-box testing*, dan *Grey-box testing*[16].

1. *White-box testing*, pengujian terfokus pada kode sistem.
2. *Black-box testing*, pengujian terhadap fungsionalitas sistem.
3. *Grey-box testing*, gabungan dari *White-box testing* dan *Black-box testing* yaitu melakukan pengujian menyeluruh terhadap fungsional dan kode pada sistem.

Berdasarkan pengujian yang akan dilakukan berkaitan dengan pengujian fungsional terhadap interaksi pengguna dan sistem, maka pengujian *Black-box testing* digunakan untuk menguji fungsionalitas dari sistem yang dikembangkan. Dengan melakukan *Black-box testing* maka dapat menemukan beberapa hal pada sistem yang dikembangkan [17].

1. Kesalahan fungsi atau fungsi tidak tersedia
2. Kesalahan pada tampilan (*interface error*)
3. Kesalahan struktur data dan akses database
4. Kesalahan performa sistem
5. Kesalahan terminasi dan inisialisasi

Pengujian fungsionalitas akan dilakukan oleh petani, programmer, produsen olahan kopi, mahasiswa, dan widyaiswara pertanian dengan 11 orang responden yaitu 2 orang programmer, 3 mahasiswa program studi Teknik pertanian, 1 owner caffe motokoffie, 1 orang widyaiswara balai pelatihan pertanian, dan 4 mahasiswa umum. Apabila kesalahan atau error mengganggu fungsionalitas aplikasi maka akan dilakukan perbaikan. Poin pengujian didasarkan apabila skor pengujian mencapai 0,5 atau mendekati 1 maka Tingkat fungsional aplikasi dikatakan baik dengan penghitungan menggunakan rumus berikut:

$$F = \frac{S}{J} \quad (1)$$

Keterangan:

F = Fungsionalitas

S = Pengujian berhasil

J = Jumlah pengujian

Untuk menguji kesesuaian sistem diperlukan pengujian *compatibility* terhadap versi

sistem operasi android. *Compatibility* adalah karakteristik pengukuran untuk mengetahui sampai sejauh mana sistem dapat bertukar informasi terhadap sistem lain dan melakukan segala fungsi yang diisyaratkan saat berbagi lingkungan perangkat lunak maupun perangkat keras yang sama [18]. Untuk pengujian *compatibility* dilakukan pada sistem operasi android 8.0 Oreo (API Level 26) sampai android 14.0 Upside Down Cake (API Level 34) menggunakan *virtual device* pada *firebase test lab*.

Tabel 1. *Device* dan versi android untuk pengujian

No	Device	Versi Android
1	Samsung Galaxy S9 (<i>Firebase test lab</i>)	Android 8.0 (Oreo)
2	Xiaomi Redmi 6A (<i>Firebase test lab</i>)	Android 8.1 (Oreo)
3	Sharp AQUOS sense2 SH-01L (<i>Firebase test lab</i>)	Android 9.0 (Pie)
4	Samsung Galaxy S20 (<i>Firebase test lab</i>)	Android 10.0 (Q)
5	Motorola G20 (<i>Firebase test lab</i>)	Android 11.0 (R)
6	Sony SO-41A (<i>Firebase test lab</i>)	Android 12.0 (S)
7	Google Pixel 6a (<i>Firebase test lab</i>)	Android 12.0L Fature Drop (S)
8	Samsung Galaxy S22 Ultra (<i>Firebase test lab</i>)	Android 13.0 (Tiramisu)
9	Google Pixel 8 (<i>Firebase test lab</i>)	Android 14.0 (Upside Down Cake)

3. Hasil dan Pembahasan

3.1. Hasil

Model tensorflow lite yang digunakan adalah model tensorflow yang telah dilakukan proses *training* model dengan hasil akurasi 90.67% dengan proses epoch sebanyak 55 kali dengan tiga kelas yaitu arabika, robusta dan liberika. Selain memuat tiga kelas utama, model juga dapat melakukan klasifikasi dengan tiga parameter berbeda yaitu daun, buah dan biji kopi. Tahapan implementasi dilakukan dengan Langkah-langkah sebagai berikut:

3.1.1. Perencanaan Syarat-syarat

Perencanaan syarat-syarat didasarkan dari studi literatur dan hasil dari wawancara. Dari tahapan yang dilakukan menghasilkan kebutuhan sistem seperti pada Tabel 2 dengan SR sebagai kode penomoran untuk kebutuhan sistem (System requirement).

Tabel 2. Kebutuhan sistem (*System requirement*)

No	Waktu Proses	Memori
1	SR-1	Sistem dapat mendeteksi jenis kopi dengan menggunakan foto daun, buah dan biji
2	SR-2	Dapat melakukan upload gambar dari galeri
3	SR-3	Dapat melakukan upload gambar dari kamera

3.1.2. Workshop Desain RAD

Dalam tahap ini dilakukan perancangan terhadap usecase diagram dan activity diagram sebagai gambaran sistem yang dikembangkan.

a. *Usecase diagram*

Usecase diagram bertujuan sebagai gambaran untuk pengguna dalam berinteraksi terhadap sistem.

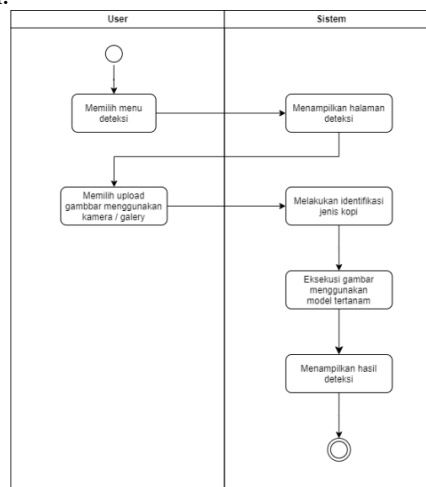


Gambar 2. *Usecase Diagram*

Dari *usecase* pada gambar 2, pengguna dapat melakukan deteksi dengan menggunakan metode upload gambar dari galeri maupun kamera secara langsung. Pengguna yang akan melakukan deteksi dapat memilih metode upload gambar dengan menggunakan kamera maupun galeri. Setelah memilih metode upload gambar dan memilih gambar maka aplikasi akan menampilkan hasil deteksi.

b. *Activity diagram*

Activity diagram digunakan untuk menunjukkan bagaimana pengguna menggunakan fitur yang terdapat pada sistem.



Gambar 3. *Activity Diagram*

Dari gambar 3, pengguna dapat melihat tahapan penggunaan aplikasi. Dalam melakukan deteksi, pengguna perlu untuk memilih menu deteksi. Setelah itu pengguna dapat menentukan metode upload gambar menggunakan kamera atau galeri. Setelah memilih gambar maka akan otomatis menampilkan hasil deteksi dari gambar yang telah dipilih.

Implementasi

Pada tahap implementasi dilakukan pengkodean aplikasi dengan menggunakan *framework* flutter dan bahasa pemrograman dart dari hasil rancangan pada tahapan workshop desain. Hasil dari pengkodean kemudian akan dilakukan pengujian dengan menggunakan metode *Black-box testing*.

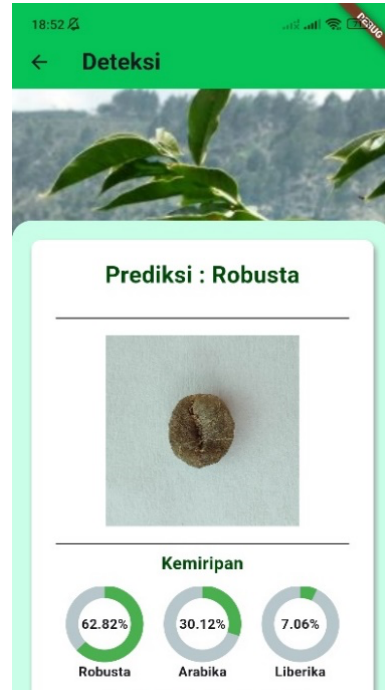
a. Pengkodean

Tahapan pengkodean dilakukan dengan menggunakan bantuan *package* dari flutter berupa *tflite_flutter* dan *tflite_flutter_helper* untuk memanggil dan melakukan interpretasi

model yang telah di import kedalam project flutter. Untuk memproses label diperlukan file dengan format *txt* yang diimport kedalam project flutter untuk memuat label kelas. Model dan label yang telah di-*import* kemudian dideklarasikan pada file *pubspec.yaml*. kemudian dilakukan penambahan metode pengambilan gambar menggunakan *image_picker* yang nantinya gambar yang di ambil dapat diproses pada model yang ditanamkan pada program dan menghasilkan persentase *confidence*. Gambar 4 dan 5 menunjukkan hasil implementasi yang telah dilakukan.



Gambar 4. Hasil Implementasi memilih model deteksi



Gambar 5. Hasil implementasi tampilan hasil deteksi

b. Pengujian

Proses pengujian menggunakan *black-box testing* dengan jumlah user penguji sebelas orang. Pengujian dilakukan oleh 2 orang programmer, 3 mahasiswa program studi Teknik pertanian, 1 owner caffe motokoffie, 1 orang widiaiswara balai pelatihan pertanian, dan 4 mahasiswa umum. Beberapa case uji terdapat pada table 2.

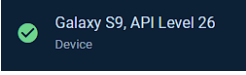
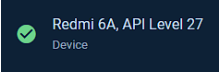
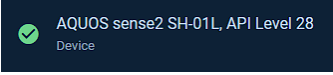
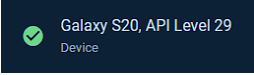
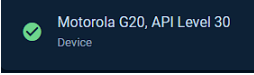
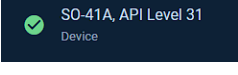
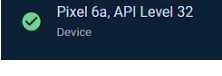
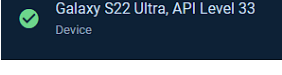
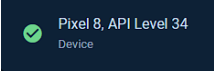
Tabel 3. Hasil pengujian *Black-box testing*

No	Test case	Sesuai	Tidak Sesuai	Total
1	Menu halaman metode ambil gambar	11		11
2	Upload gambar menggunakan galeri	11		11
3	Upload Gambar Menggunakan Kamera	10	1	11
4	Hasil Deteksi	11		11
	Total	43	1	44

Dari hasil pengujian tersebut terdapat satu test case yang tidak sesuai. Dari hal tersebut terdapat catatan dari penguji aplikasi bahwa fitur tetap berjalan namun proses terjadi delay saat pengambilan gambar menggunakan kamera. Sehingga ketidak sesuaian tersebut tidak mempengaruhi fungsionalitas dari aplikasi, namun terdapat pengaruh pada performa aplikasi.

Pengujian *compatibility* dilakukan menggunakan *firebase test lab*. Pengujian ini dilakukan untuk mengetahui kesesuaian aplikasi pada setiap versi sistem operasi yang berbeda. Tabel 4 merupakan hasil pengujian *compatibility* dari sistem yang dikembangkan.

Tabel 3. Hasil pengujian *Black-box testing*

No	Device	Versi Android	Hasil	Proses
1	Samsung Galaxy S9 (<i>Firebase test lab</i>)	Android 8.0 (Oreo)		Berhasil
2	Xiaomi Redmi 6A (<i>Firebase test lab</i>)	Android 8.1 (Oreo)		Berhasil
3	Sharp AQUOS sense2 SH-01L (<i>Firebase test lab</i>)	Android 9.0 (Pie)		Berhasil
4	Samsung Galaxy S20 (<i>Firebase test lab</i>)	Android 10.0 (Q)		Berhasil
5	Motorola G20 (<i>Firebase test lab</i>)	Android 11.0 (R)		Berhasil
6	Sony SO-41A (<i>Firebase test lab</i>)	Android 12.0 (S)		Berhasil
7	Google Pixel 6a (<i>Firebase test lab</i>)	Android 12.0L Feture Drop (S)		Berhasil
8	Samsung Galaxy S22 Ultra (<i>Firebase test lab</i>)	Android 13.0 (Tiramisu)		Berhasil
9	Google Pixel 8 (<i>Firebase test lab</i>)	Android 14.0 (Upside Down Cake)		Berhasil

3.2. Pembahasan

Dari penelitian yang dilakukan menghasilkan aplikasi android berbasis flutter. Dengan menggunakan metode pengembangan perangkat lunak RAD mebantukan dalam menyelesaikan proses pengembangan sistem dengan waktu yang lebih efisien. Metode pengujian yang dilakukan dapat mengetahui berbagai kesalahan dan error yang terjadi pada sistem yang dikembangkan berbasis pada fungsional sistem.

Dengan adanya *framework* flutter dan bahasa pemrograman dart memudahkan dalam pengembangan aplikasi. *Package* yang disediakan flutter dapat membantu pengerjaan lebih

cepat dan memudahkan proses implementasi model tensorflow lite. model tensorflow lite yang tertanam dalam aplikasi membantu mempermudah proses deteksi dan lebih ringan.

Dari pengujian yang telah dilakukan menggunakan *black-box testing*. Dapat dilakukan penghitungan skor kelayakan menggunakan rumus (1).

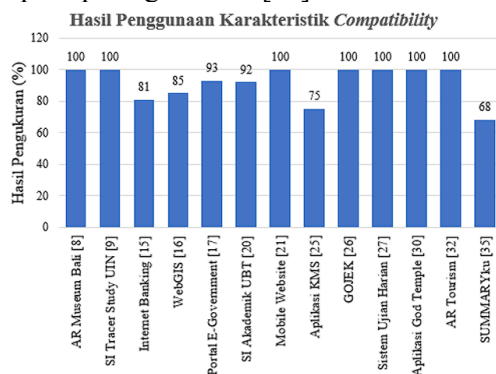
$$F = \frac{S}{J}$$

$$F = \frac{43}{44}$$

$$F = 0.97$$

Dari hasil penghitungan pengujian tersebut maka aplikasi telah dikatakan layak secara fungsionalitas dengan skor kelayakan 0.96. hal tersebut telah memenuhi standar skor yaitu 0.5 hingga mendekati atau sama dengan 1.0.

Hasil pengujian *compatibility* menunjukkan 100% kesesuaian aplikasi yang dikembangkan sesuai dengan semua versi sistem operasi android yang ditentukan dalam pengembangan sistem. Dengan versi android 8.0 Oreo (API Level 26) sampai android 14.0 Upside Down Cake (API Level 34). Hal ini menunjukkan aplikasi dapat berjalan pada perangkat android dengan sistem operasi android 8.0 hingga android 14.0. Dengan perbandingan pada penelitian yang dilakukan Mulyawan pada tahun 2021 menunjukkan hasil uji *compatibility* pada beberapa aplikasi android seperti pada gambar 6 [18].



Gambar 6. Perbandingan pengukuran *Compatibility* pada penelitian Mulyawan, et. Al [18]

4. Kesimpulan

Dari penelitian yang dilakukan menghasilkan sebuah aplikasi pendeteksi jenis kopi dengan input parameter berupa biji, buah dan daun. Aplikasi ini dapat berguna bagi Masyarakat yang memiliki kebingungan dalam membedakan jenis kopi berdasarkan karakteristik fisik dari tanaman kopi. Dengan hasil pengujian fungsionalitas dengan 11 perangkat android berbeda yang dilakukan oleh satu orang widyaiswara balai pelatihan pertanian, satu orang owner caffe, empat mahasiswa umum, empat mahasiswa teknik pertanian dan dua orang programmer, aplikasi memperoleh skor fungsionalitas mencapai 0.97 yang menyatakan bahwa aplikasi telah mendekati fungsionalitas yang hampir sempurna dari keseluruhan fungsi pada aplikasi yang dibangun. Hasil pengujian *compatibility* menunjukkan bahwa aplikasi dapat berjalan dengan baik pada perangkat android dengan sistem operasi android 8.0 sampai android 14.0 yang menyatakan bahwa aplikasi dapat bertukar informasi pada seluruh perangkat pada sistem operasi android yang telah ditentukan sehingga semakin banyak sistem versi sistem operasi android yang dapat menjalankan aplikasi deteksi jenis kopi maka akan semakin banyak pengguna yang dapat menjalankan aplikasi tersebut. Dengan berhasilnya dibangunnya aplikasi deteksi jenis kopi, aplikasi ini dapat berguna dalam membantu proses identifikasi jenis kopi

<https://doi.org/10.31849/digitalzone.v15i1.19563>

Digital Zone is licensed under a Creative Commons Attribution International (CC BY-SA 4.0)

sehingga dapat mengurangi potensi kesalahan dalam melakukan identifikasi jenis kopi. Hal tersebut memungkinkan berkurangnya potensi terjadinya miskomunikasi supplier bibit, buah, maupun biji terhadap petani, konsumen maupun produsen olahan kopi dengan melakukan deteksi terlebih dahulu menggunakan aplikasi deteksi jenis kopi.

Daftar Pustaka

- [1] M. Ibnu and N. Rosanti, "Tren Produksi Dan Perdagangan Negara-Negara Produsen Kopi Terbesar Di Dunia Dan Implikasinya Bagi Indonesia," *Buletin Ilmiah Litbang Perdagangan*, vol. 16, no. 2, Dec. 2022, doi: [10.55981/bilp.2022.5](https://doi.org/10.55981/bilp.2022.5).
- [2] A. Susanto, Y. Kusumawati, E. D. Niagara, and C. A. Sari, "Convolutional Neural Network Dalam Sistem Deteksi Helm Pada Pengendara Motor," *Seminar Nasional Teknologi dan Multidisiplin Ilmu (SEMNASTEKMU)*, vol. 2, no. 1, pp. 91–99, Dec. 2022, doi: [10.51903/semnastekmu.v2i1.158](https://doi.org/10.51903/semnastekmu.v2i1.158).
- [3] S. Garg and N. Baliyan, "Comparative analysis of Android and iOS from security viewpoint," *Comput Sci Rev*, vol. 40, p. 100372, May 2021, doi: [10.1016/j.cosrev.2021.100372](https://doi.org/10.1016/j.cosrev.2021.100372).
- [4] M. Fachriyan and D. Dharmayanti, "Pembangunan Aplikasi Pengenalan Objek Terdekat Untuk Penyandang Tunanetra Menggunakan Mlkit Dan Text To Voice Berbasis Android," *elibrary UNIKOM*, Jul. 2020. <http://dx.doi.org/10.14710/transient.v10i3.510-516>
- [5] R. Arianto, "Perancangan Aplikasi Pengenalan dan Pembelajaran Pertanian Milenial Hidroponik Berbasis Android Flutter & Dart Dengan Metode RAD," *OKTAL : Jurnal Ilmu Komputer dan Science*, vol. 2, no. 3, pp. 928–935, Mar. 2023.
- [6] A. Setiawan, A. Irma Purnamasari, and N. Suarna, "Penerapan Rapid Application Development Melalui Sistem Informasi Geografis Untuk Pemetaan Lembaga Pendidikan Anak Usia Dini," *Infotekmesin*, vol. 10, no. 1, pp. 23–27, Jan. 2019, doi: [10.35970/infotekmesin.v10i1.23](https://doi.org/10.35970/infotekmesin.v10i1.23).
- [7] A. Noertjahyana, "Studi Analisis Rapid Application Development Sebagai Salah Satu Alternatif Metode Pengembangan Perangkat Lunak," *Jurnal Informatika*, vol. 3, pp. 74–79, Nov. 2022, doi: [10.9744/informatika.3.2.pp.%2064-68](https://doi.org/10.9744/informatika.3.2.pp.%2064-68).
- [8] D. Manajang, S. R. U. A. Sompie, and A. Jacobus, "Implementasi Framework Tensorflow Object Detection API Dalam Mengklasifikasi Jenis Kendaraan Bermotor," *Jurnal Teknik Informatika*, vol. 15, no. 3, pp. 171–178, Feb. 2021. <https://doi.org/10.35793/jti.15.3.2020.29775>
- [9] M. B. Baihaqi, Y. Litanianda, and A. Triyanto, "Implementasi Tensor Flow Lite Pada Teachable Untuk Identifikasi Tanaman Aglonema Berbasis Android," *Komputek*, vol. 6, no. 1, p. 70, Apr. 2022, doi: [10.24269/jkt.v6i1.1143](https://doi.org/10.24269/jkt.v6i1.1143).
- [10] R. 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, Sep. 2022, doi: [10.17529/jre.v18i3.8754](https://doi.org/10.17529/jre.v18i3.8754).
- [11] B. Raharjo, *Pemrograman Android Dengan Flutter*. 2019.
- [12] M. Y. Putra and D. E. Kurniawan, "Implementasi Sistem Reminder Jadwal pada eLearning Moodle Berbasis API Menggunakan Framework Flutter," *Journal of Applied Computer Science and Technology*, vol. 4, no. 1, pp. 7–11, Jun. 2023, doi: [10.52158/jacost.v4i1.490](https://doi.org/10.52158/jacost.v4i1.490).
- [13] R. Mufidianto, J. S. Asri, N. Anwar, and T. I. Widyawan, "Perancangan Aplikasi Antrean Makanan Berbasis Cross-Platform Dengan Framework Flutter (Studi Kasus Rm. Padang Buaran)," *IKRAITH-TEKNOLOGI*, vol. 7, no. 3, pp. 67–75, Nov. 2023, <https://doi.org/10.37817/ikraith-teknologi.v7i3.3235>

- [14] T. M. Pratama and A. R. Pratama, “Rancang Bangun Aplikasi Android ‘Kuliah Apa?’ Berbasis Flutter dan TensorFlow Lite,” *AUTOMATA*, vol. 2, no. 1, Jan. 2021.
- [15] V. Febrian, M. R. Ramadhan, M. Faisal, and A. Saifudin, “Pengujian pada Aplikasi Penggajian Pegawai dengan menggunakan Metode Blackbox,” vol. 5, no. 1, pp. 2622–4615, 2020, [Online]. <https://doi.org/10.32493/informatika.v5i1.4340>
- [16] I. R. Dhaifullah, M. Muttanifudin H, A. Ananda Salsabila, and M. Ainul Yaqin, “Survei Teknik Pengujian Software,” *Journal Automation Computer Information System*, vol. 2, no. 1, pp. 31–38, Jun. 2022, [doi: 10.47134/jacis.v2i1.42](https://doi.org/10.47134/jacis.v2i1.42).
- [17] M. Mursyidah and H. T. Hidayat, “Pengujian Sistem Informasi Akuntansi Biaya Operasional Sekolah Dengan Black Box Testing,” *Jurnal Infomedia*, vol. 2, no. 2, pp. 7–14, Jan. 2018, [doi: 10.30811/v2i2.512](https://doi.org/10.30811/v2i2.512).
- [18] M. D. Mulyawan, I. N. S. Kumara, I. B. A. Swamardika, and K. O. Saputra, “Kualitas Sistem Informasi Berdasarkan ISO/IEC 25010: Literature Review,” *Majalah Ilmiah Teknologi Elektro*, vol. 20, no. 1, p. 15, Mar. 2021, [doi: 10.24843/mite.2021.v20i01.p02](https://doi.org/10.24843/mite.2021.v20i01.p02).