

**APLIKASI PENGENALAN JENIS BIJI KOPI
BERDASARKAN GREEN BEAN BERBASIS ANDROID DAN
WEBSITE**

Tio Dwi Satrio¹, Donny Avianto²

1,2) Informatika, Sains dan Teknologi, Universitas Teknologi Yogyakarta, Indonesia

Article Info	ABSTRACT
Article history: Received: 05 November 2024 Revised: 07 November 2024 Accepted: 07 November 2024	Abstrak Penelitian ini bertujuan untuk mengembangkan aplikasi klasifikasi biji kopi arabika, robusta, dan excelsa menggunakan model <i>MobileNet</i> dan algoritma <i>Convolutional Neural Network</i> (CNN). Metode CNN dipilih karena kemampuannya mengekstraksi fitur visual dari gambar dengan akurasi tinggi. Penelitian ini menggunakan data berupa ratusan gambar biji kopi yang dikategorikan berdasarkan jenisnya, seperti arabika, robusta, dan excelsa. Proses pelatihan model dilakukan dengan menggunakan teknik augmentasi data untuk memperluas variasi dataset dan mengurangi kemungkinan <i>overfitting</i>. Hasil pengujian menunjukkan bahwa <i>MobileNet</i> dapat diterapkan untuk klasifikasi pada perangkat Android. Sektor kopi diharapkan akan mendapat manfaat besar dari aplikasi ini, terutama di bidang mengidentifikasi jenis biji kopi yang akan memungkinkan petani, pedagang, dan <i>roaster</i> untuk dengan cepat mengenali berbagai jenis biji kopi. Kajian ini diharapkan dapat memberikan wawasan secara detail mengenai penerapan teknologi <i>machine learning</i>, khususnya <i>deep learning</i>, pada sektor perkebunan, khususnya pada klasifikasi produk perkebunan, yang berpotensi meningkatkan kualitas dan produktivitas secara keseluruhan. Kata Kunci: Aplikasi, <i>Convolutional Neural Network</i>, Kopi Abstract <i>This research aims to develop an application for classifying arabica, robusta, and excelsa coffee beans using the MobileNet model and Convolutional Neural Network (CNN) algorithms. The CNN method was chosen for its ability to extract visual features from images with high accuracy. This study utilizes data comprising hundreds of images of coffee beans categorized by type, such as arabica, robusta, and excelsa. The model training process employed data augmentation techniques to expand the dataset's variability and reduce the likelihood of overfitting. The test results indicate that MobileNet can be applied for classification on Android devices. The coffee sector is expected to benefit significantly from this application, particularly in identifying coffee bean types, allowing farmers, traders, and roasters to quickly recognize various coffee bean varieties. This study is anticipated to provide detailed insights into the application of machine learning technology, especially deep learning, in the plantation sector, specifically in the classification of agricultural products, which has the potential to enhance quality and productivity.</i> Keywords: Application, <i>Convolutional Neural Network</i>, Coffee

Djtechno: Jurnal Teknologi Informasi oleh Universitas Dharmawangsa Artikel ini bersifat open access yang didistribusikan di bawah syarat dan ketentuan dengan Lisensi Internasional Creative Commons Attribution NonCommercial ShareAlike 4.0 ([CC-BY-NC-SA](https://creativecommons.org/licenses/by-nc-sa/4.0/)).



Corresponding Author:

E-mail : tiodwisatrio27@gmail.com

1. PENDAHULUAN

Kopi merupakan produk perkebunan yang diminati dan mempunyai nilai pasar yang besar. (Hasan, 2022). Di seluruh dunia, kopi tidak hanya menjadi minuman populer, tetapi juga merupakan bagian penting dari budaya dan kehidupan sosial (Gumulya & Helmi, n.d.). Berbagai jenis biji kopi, seperti arabika, robusta, dan excelsa, memiliki rasa, aroma, dan kualitas yang berbeda-beda. Setiap jenis biji kopi memiliki karakteristik unik yang mempengaruhi pengalaman dalam menyeduh dan menikmati kopi. Namun, seiring dengan meningkatnya jenis biji kopi yang tersedia di pasaran, tantangan untuk mengidentifikasi dan mengklasifikasikan biji kopi secara akurat menjadi semakin kompleks.

Mengidentifikasi jenis biji kopi berdasarkan *green bean* dengan benar sangat penting untuk menjaga kualitas produk dan memenuhi preferensi konsumen (Nugroho & Sebatubun, 2020). Berbeda dari penelitian terdahulu yang berfokus pada klasifikasi tingkat pemanggangan (*profile roasting*). Penelitian identifikasi *profile roasting* sering kali tidak memperhatikan jenis kopi tertentu karena fitur visual, seperti warna dan tekstur menjadi serupa setelah pemanggangan. Sebaliknya, *green bean* memiliki karakteristik fisik unik sesuai jenisnya, yang memungkinkan identifikasi lebih akurat sebelum pemanggangan. Pendekatan ini penting untuk memastikan kualitas kopi sejak awal rantai pasok dan memungkinkan penyesuaian *profile roasting* yang optimal.

Kemajuan teknologi pemrosesan gambar dan kecerdasan buatan telah membuka jalan baru dalam berbagai bidang, termasuk pertanian (Anu et al., n.d.). Kecerdasan buatan adalah cabang ilmu komputer yang berfokus pada pengembangan sistem yang

mampu melakukan tugas yang biasanya memerlukan kecerdasan manusia, seperti pengenalan pola, pengambilan keputusan, dan pembelajaran dari data.

Salah satu pendekatan yang menunjukkan hasil menjanjikan dalam penggunaan kecerdasan buatan adalah *deep learning*, yang merupakan bagian dari *machine learning*. *Deep learning* memproses data menggunakan jaringan saraf tiruan yang terdiri dari banyak lapisan (Zhu et al., 2018). Di antara berbagai arsitektur *deep learning*, *Convolutional Neural Network* (CNN) paling umum digunakan untuk pemrosesan gambar (Elngar et al., 2021). CNN dirancang khusus untuk mendeteksi pola dalam data gambar dengan mengekstraksi fitur-fitur penting melalui serangkaian lapisan konvolusional, *pooling*, dan *fully-connected* (Suwitono & Kaunang, 2022).

Convolutional Neural Network (CNN) adalah jenis jaringan saraf tiruan yang dibangun untuk memproses data dalam topologi mirip *grid*, seperti gambar. CNN menggunakan *layer convolutional* untuk mengekstrak fitur penting dari gambar, lalu menggabungkan lapisan tersebut untuk mengurangi dimensi data dan mempercepat komputasi (Thoma, 2017). Dengan menggunakan arsitektur *multilayer*, CNN dapat mempelajari hierarki fitur dari data yang kompleks, memungkinkan model mengenali pola dan objek dalam gambar dengan akurasi tinggi. Sejak diperkenalkan, CNN telah menjadi standar untuk berbagai aplikasi pemrosesan gambar seperti klasifikasi gambar, deteksi objek, dan segmentasi gambar. CNN dapat mencapai kinerja unggul dalam kontes klasifikasi gambar sehingga minat untuk menggunakan teknik ini dalam bidang *computer vision* semakin meningkat (Michele et al., 2019). Selain itu, perbaikan dalam arsitektur dan teknik pelatihan membuat CNN semakin efisien dan efektif dalam aplikasi dunia nyata, seperti pengenalan wajah dan klasifikasi objek dalam konteks yang lebih kompleks (Tayal et al., 2022).

MobileNet adalah arsitektur jaringan saraf tiruan yang dikembangkan oleh Google untuk memenuhi kebutuhan aplikasi *deep learning* pada perangkat dengan sumber daya terbatas seperti ponsel dan perangkat *Internet of Things* (IoT) (Saputra et al., 2021). *MobileNet* menggunakan struktur konvolusional yang dapat dipisahkan secara mendalam (Sulistio et al., 2023). Ini adalah teknik konvolusi yang lebih efisien dibandingkan konvolusi tradisional di CNN (*Convolutional Neural Network*) (Aini &

Liliana, 2022). Dengan membagi konvolusi menjadi dua langkah, yaitu kedalaman dan titik. *MobileNet* dapat secara signifikan mengurangi jumlah parameter dan operasi komputasi, sehingga lebih menghemat daya dan memori.

Arsitektur *MobileNet* memperkenalkan dua *hyperparameter* utama: pengganda lebar dan pengganda resolusi. Pengganda lebar mengurangi ukuran model dengan mengurangi jumlah fitur di setiap lapisan, sedangkan pengganda resolusi mengurangi resolusi gambar masukan, memungkinkan pengguna mengontrol keseimbangan antara akurasi dan kecepatan prediksi tergantung pada kebutuhan perangkat (Hardi, 2022). *MobileNet* dapat digunakan untuk tugas deteksi objek dan klasifikasi gambar pada perangkat seluler tanpa memerlukan perangkat keras dalam jumlah besar, menjadikannya model populer untuk aplikasi yang memerlukan kecepatan dan efisiensi, seperti deteksi objek, klasifikasi tumbuhan, dan pengenalan wajah (Sulkifly Said et al., 2024).

Kotlin adalah bahasa pemrograman modern yang dikembangkan oleh JetBrains yang bertujuan untuk menyederhanakan proses pengembangan aplikasi dan menyediakan *syntax* yang bersih, intuitif, dan mudah dipahami. Kotlin pertama kali diperkenalkan pada tahun 2011 dan mendapat dukungan resmi dari Google pada tahun 2017 sebagai bahasa pemrograman utama untuk pengembangan Android. Karena Kotlin dikompilasi ke *bytecode* Java, Kotlin dapat berjalan di Java Virtual Machine (JVM) dan berinteraksi langsung dengan kode Java (Schwermer, 2018).

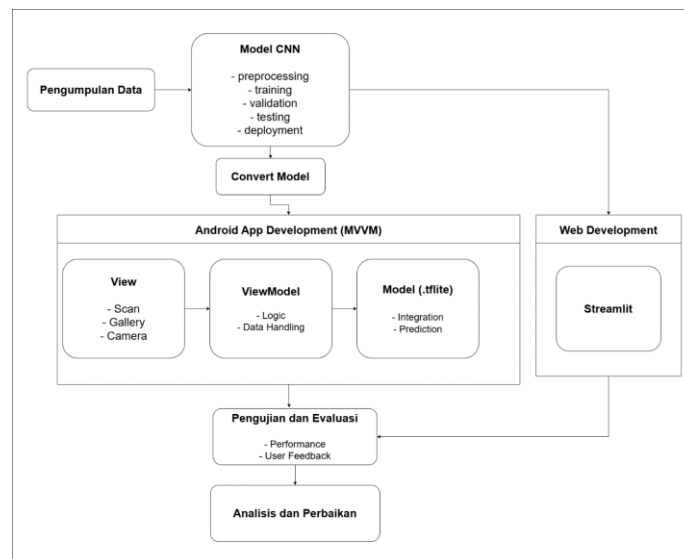
Penelitian ini bertujuan untuk mengembangkan aplikasi berbasis CNN yang dapat secara otomatis mengklasifikasikan jenis biji kopi berdasarkan gambar. Melalui proses pelatihan model dengan dataset gambar biji kopi yang beragam, diharapkan aplikasi ini mampu mengenali dan membedakan antara berbagai jenis biji kopi dengan tingkat akurasi yang tinggi. Selain itu, penelitian ini juga akan membahas tahapan pengumpulan data, *preprocessing* citra, dan pengoptimalan model CNN untuk mencapai performa terbaik.

Aplikasi yang dihasilkan diharapkan dapat digunakan oleh petani, pedagang, *roaster* dan konsumen untuk meningkatkan proses identifikasi biji kopi. Dengan

pemahaman yang lebih baik tentang jenis biji kopi, diharapkan para pelaku industri kopi dapat mengambil keputusan yang lebih baik pada tahap pemrosesan kopi selanjutnya, seperti *roasting* dan *brewing*, yang pada akhirnya akan berkontribusi pada pertumbuhan dan keberlanjutan sektor kopi secara keseluruhan. Dengan demikian, penelitian ini tidak hanya berkontribusi pada pengembangan teknologi informasi dan kecerdasan buatan, tetapi juga berpotensi meningkatkan kualitas produk kopi serta daya saing kopi di pasar *global*.

2. METODE PENELITIAN

Tahap-tahap penelitian yang ditempuh untuk mengembangkan aplikasi pada penelitian ini dapat dilihat pada Gambar 1.



Gambar 1. Tahapan Penelitian

Adapun penjelasan detail dari tahapan penelitian Gambar 1 adalah sebagai berikut:.

a. Pengumpulan Data

Proses ini dilakukan dengan mengambil gambar langsung dari petani kopi di daerah Magelang. Peneliti mulai dengan mengidentifikasi petani kopi yang bersedia berkontribusi dan menyediakan akses ke biji kopi mereka. Gambar-gambar diambil menggunakan kamera dari smartphone dengan menggunakan fitur makro untuk memastikan setiap gambar representatif, dengan pengambilan gambar dari berbagai sudut dan kondisi pencahayaan.

Setiap gambar kemudian diberi label sesuai dengan jenis biji kopi yang dikategorikan menjadi arabika, robusta, atau excelsa. Semua gambar yang dikumpulkan diorganisir dalam folder terpisah berdasarkan jenisnya dan disimpan dalam format JPEG untuk memudahkan pengolahan lebih lanjut. Pada penelitian ini digunakan total 750 data, dengan 250 data untuk setiap jenis biji kopi arabika, robusta, dan excelsa.

b. Model CNN

Setelah data terkumpul, penulis kemudian melakukan pembuatan model dengan langkah-langkah berikut:

1. *Preprocessing*

Pada tahap *preprocessing*, penulis melakukan *resize*, yaitu mengubah ukuran gambar biji kopi menjadi 224×224 pixel agar sesuai dengan ukuran *input* standar *MobileNet*. Selanjutnya, penulis melakukan normalisasi dengan mengubah nilai *pixel* ke rentang $[0, 1]$ untuk mempermudah konvergensi model saat pelatihan. Untuk mencegah *overfitting*, penulis melakukan augmentasi data dengan menambah variasi data menggunakan teknik *rotation*, *zoom*, dan *flipping*. *Rotation* diterapkan dalam rentang -30° hingga $+30^\circ$ (*rotation_range=30*), memungkinkan gambar dapat diputar secara acak. *Zoom* digunakan dalam rentang 0.7 hingga 1.2 kali (*zoom_range = [0.7, 1.2]*) ukuran asli untuk memberikan variasi dalam ukuran objek kopi. *Horizontal Flipping* (*horizontal_flip=True*) dan *Vertical Flipping* (*vertical_flip=True*) juga diterapkan secara acak agar membantu model mengenali kopi dari orientasi kiri ke kanan dan atas ke bawah. Dengan teknik ini, model menjadi lebih *adaptive* terhadap berbagai posisi, ukuran, dan orientasi gambar kopi.

2. *Training*

Pada tahap *training*, penulis melakukan inisialisasi *MobileNet* untuk memuat model dengan bobot awal dari *ImageNet*, tetapi tanpa

lapisan akhir untuk menyesuaikan klasifikasi tiga kelas (arabika, robusta, dan excelsa). Selanjutnya, penulis menerapkan *transfer learning* dengan melatih ulang lapisan atas agar dapat mengenali biji kopi, sementara lapisan bawah dipertahankan untuk mendeteksi fitur dasar.

3. *Testing*

Setiap gambar dalam data uji dimasukkan ke dalam model untuk diprediksi. Model memberikan output berupa probabilitas untuk masing-masing kelas (arabika, robusta, excelsa), dan kelas dengan probabilitas tertinggi dipilih sebagai prediksi model untuk gambar tersebut.

4. *Evaluation*

Kemudian penulis melakukan evaluasi terhadap model untuk mengukur dan memahami kinerja model dengan lebih mendalam, Metriks evaluasi yang digunakan antara lain: *confussion matrix*, akurasi, *Precision*, *Recall*, dan *F1-score*.

5. *Deployment*

Terakhir, penulis melakukan *deployment* agar model dapat digunakan secara lokal dalam aplikasi Android.

c. *Convert Model*

Penulis melakukan konversi terhadap model *MobileNet* ke dalam format *.tflite* agar dapat diintegrasikan dengan aplikasi Android menggunakan *Tensorflow Lite Interpreter*.

d. *Arsitektur MVVM*

Penulis memilih arsitektur MVVM (*Model*, *View*, *ViewModel*) agar menjaga keterpisahan antara logika UI, data, dan pengolahan data (Maulana et al., 2022). Berikut penjelasan pada masing-masing *layer* pada arsitektur MVVM tersebut.

1. *View Layer*

Layer ini merupakan antarmuka pengguna aplikasi, di mana interaksi pengguna berlangsung (Maulana et al., 2022). Dalam konteks penelitian

ini, *View Layer* mencakup komponen seperti *Scan* dan *Gallery*. Fitur *scan* berfungsi untuk mengambil gambar biji kopi menggunakan kamera perangkat. Sedangkan fitur *gallery* memungkinkan pengguna untuk mengunggah gambar biji kopi dari *gallery*. Lapisan ini dirancang untuk memberikan pengalaman pengguna yang intuitif, memungkinkan pengguna untuk dengan mudah memilih atau mengambil gambar untuk klasifikasi.

2. *ViewModel Layer*

ViewModel Layer berfungsi sebagai jembatan antara *View* dan *Model* (Diantoni et al., 2024). Tugas utamanya adalah menangani logika aplikasi dan mengelola data. Dalam kasus ini, *ViewModel* melakukan beberapa fungsi penting, seperti Pemrosesan Gambar dan Pemanggilan Model. Pada tahap Pemrosesan Gambar, *ViewModel* akan melakukan *preprocessing* gambar untuk memastikan gambar yang diunggah oleh pengguna sesuai dengan format yang diperlukan oleh model. Kemudian, pada tahap Pemanggilan Model, *ViewModel* akan berinteraksi dengan model untuk melakukan prediksi berdasarkan gambar yang telah diproses. *ViewModel* juga akan mengelola hasil yang diterima dari model untuk ditampilkan kembali ke pengguna. Dengan memisahkan logika antarmuka dari logika bisnis, lapisan ini memungkinkan pengujian dan pemeliharaan yang lebih baik (Riyadhi et al., 2023).

3. *Model Layer*

Model Layer adalah inti dari aplikasi yang bertanggung jawab untuk pengolahan data dan prediksi (Royand et al., 2024). Dalam konteks klasifikasi jenis biji kopi, *Model Layer* mencakup Integrasi Model dan Prediksi. Pada proses Integrasi Model, model yang telah dilatih akan disimpan dalam format TensorFlow Lite agar dapat dijalankan di perangkat Android. Kemudian pada proses Prediksi, *Model Layer* akan melakukan inferensi menggunakan *TensorFlow Lite Interpreter*. Hasil

prediksi ini adalah klasifikasi jenis biji kopi yang akan dikembalikan ke *ViewModel*.

e. Streamlit

Arsitektur aplikasi Streamlit untuk klasifikasi jenis biji kopi berfokus pada pemrosesan gambar dan inferensi menggunakan model CNN. Aplikasi ini dimulai dengan antarmuka pengguna yang sederhana di mana pengguna dapat mengunggah gambar biji kopi. Setelah gambar diunggah, aplikasi akan melakukan *preprocessing* untuk memastikan gambar sesuai dengan format yang dibutuhkan oleh model, termasuk mengubah ukuran gambar (224x224 *pixel*) dan menormalisasikannya. Setelah *preprocessing*, gambar tersebut akan diproses melalui model CNN yang telah dilatih, seperti *MobileNet*. Model ini akan melakukan inferensi untuk menentukan jenis biji kopi, apakah itu arabika, robusta, atau excelsa. Hasil klasifikasi yang dihasilkan oleh model kemudian akan disajikan kepada pengguna dalam bentuk label yang jelas, disertai dengan probabilitas yang menunjukkan tingkat kepercayaan model terhadap klasifikasi yang diberikan. Selain Streamlit, halaman ini juga menggunakan pustaka TensorFlow untuk pemodelan CNN dan serta PIL untuk pengolahan gambarnya.

f. Pengujian

Pengujian melibatkan pengukuran performa aplikasi, seperti waktu pemrosesan gambar dari input hingga output klasifikasi, serta stabilitas aplikasi pada berbagai perangkat Android. Untuk memastikan kualitas dan stabilitas aplikasi, pengujian dilakukan menggunakan perangkat dengan spesifikasi berbeda. Berikut perangkat yang digunakan dalam proses pengujian.

Tabel 1. Perangkat Pengujian

Nama Perangkat	RAM	Processor	Penyimpanan	Kamera	Sistem Operasi
Samsung Galaxy A54	8 GB	Exynos 1380	256 GB	50 MP, f/1.8	Android 13

Infinix Zero 5G	8 GB	MediaTek Dimensity 900	128 GB	48 MP, f/1.8	Android 11
OPPO A15S	4 GB	MediaTek Helio P35	64 GB	13 MP, f/2.2	Android 10

3. HASIL DAN PEMBAHASAN

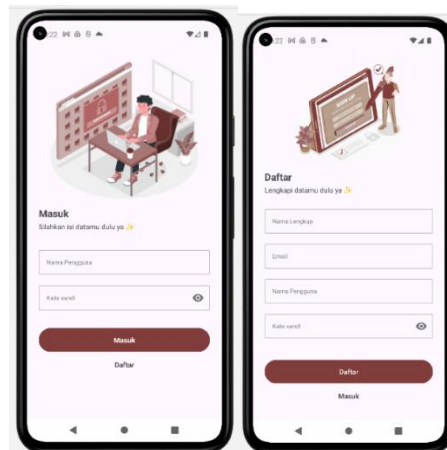
Penggunaan model *MobileNet* dalam aplikasi klasifikasi biji kopi terbukti efektif berkat arsitekturnya yang ringan dan efisien, yang dirancang khusus untuk perangkat *mobile*. Akurasi yang tinggi menunjukkan bahwa model ini mampu mengenali berbagai jenis biji kopi dengan baik, meskipun ada ruang untuk perbaikan, terutama pada jenis biji kopi yang mirip. Berikut hasil dari pengujian aplikasi pada berbagai perangkat dengan percobaan sebanyak 10 kali.

Tabel 2. Hasil Pengujian Aplikasi

No	Spesifikasi	Waktu Pengambilan Gambar (detik)	Waktu Klasifikasi (detik)	Total Waktu (detik)	Jumlah Crash	Pengalaman Pengguna
1.	Samsung Galaxy A54	1.4	0.4	1.8	0	Sangat stabil, responsif
2.	Infinix Zero 5G 2023	1.5	0.6	2.1	1	Stabil, sedikit lag pada pencahayaan minim
3.	OPPO A15 S	2.1	0.9	3.0	3	Kurang stabil, sering terjadi lag

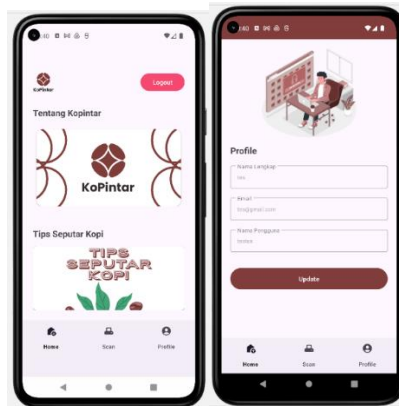
Dari tabel di atas, dapat diketahui bahwa perangkat dengan spesifikasi tinggi memiliki waktu pemrosesan yang relatif cepat. Hal ini menunjukkan bahwa *MobileNet* mampu bekerja dengan baik dalam situasi yang memerlukan kecepatan. Namun, untuk perangkat dengan spesifikasi lebih rendah, waktu pemrosesan yang lebih lama dapat mengganggu pengalaman pengguna. Hal ini menandakan perlunya optimasi lebih lanjut, seperti pengurangan resolusi gambar input atau penerapan teknik pemrosesan gambar.

Pengalaman pengguna mengindikasikan bahwa meskipun aplikasi sudah berjalan dengan baik, ada tantangan dalam kinerja pada perangkat dengan spesifikasi yang lebih rendah. Oleh karena itu, rekomendasi perbaikan meliputi penyederhanaan resolusi gambar dan penyesuaian parameter model untuk meningkatkan performa model. Secara keseluruhan, hasil pengujian menunjukkan bahwa aplikasi ini memiliki potensi besar, tetapi perlu penyesuaian untuk memastikan pengalaman pengguna yang konsisten di semua perangkat.



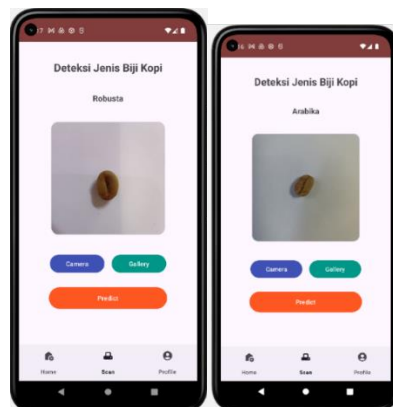
Gambar 2. Tampilan Login Register

Pengguna yang akan menggunakan aplikasi akan diarahkan ke halaman *login*. Jika pengguna belum memiliki akun, pengguna bisa mendaftar pada halaman *register* dengan mengisi data pada *form* yang ada di halaman tersebut. Kemudian data yang sudah dimasukkan di dalam halaman *register* akan disimpan di database menggunakan Firebase. Jika sudah terdaftar, pengguna bisa memasukkan data *username* dan *password* pada halaman *login*. Firebase dipilih karena kemudahannya dalam mengelola autentikasi dan integrasi backend tanpa perlu server tambahan. Dengan Firebase, data pengguna disimpan secara aman dan memungkinkan sinkronisasi real-time yang efisien, cocok untuk aplikasi mobile yang membutuhkan performa tinggi dan kecepatan akses (Khawas & Shah, 2018). Selain itu, Firebase menawarkan keamanan bawaan dan dukungan untuk autentikasi berbagai metode, seperti email, Google *Sign-In*, atau nomor telepon, sehingga memudahkan pengguna untuk masuk dengan cara yang paling nyaman bagi mereka.

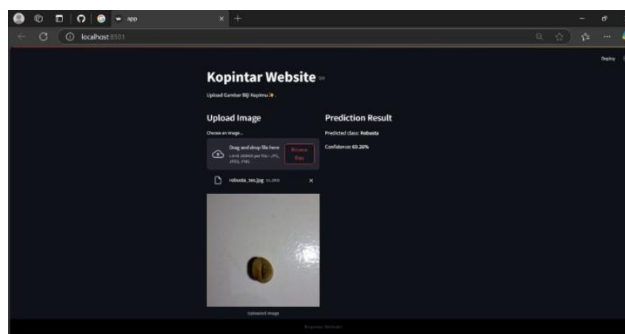


Gambar 3. Tampilan Home dan Profile

Pengguna yang sudah berhasil *login*, akan diarahkan ke halaman utama. Halaman utama memiliki *bottom navigation* yang bernavigasi ke halaman *scan* dan halaman *profile*. Di dalam halaman *profile*, pengguna bisa melihat data diri yang telah didaftarkan di halaman *register*. Pengguna juga bisa mengubah data yang telah terdaftar sendiri dengan fitur *update* yang ada pada aplikasi tersebut.



Gambar 4. Tampilan Scan Android



Gambar 5. Tampilan Hasil Website

Berdasarkan Gambar 4, dapat diketahui, model bekerja dengan cukup baik pada platform Android dengan memprediksi setiap jenis biji kopi yang diunggah oleh pengguna. Model mampu memprediksi setiap gambar dan memberikan label yang sesuai jenis kopi yang ada. Hasil yang sama juga ditunjukkan pada aplikasi berbasis web seperti yang terlihat pada Gambar 5.

4. SIMPULAN

Aplikasi untuk klasifikasi jenis biji kopi berdasarkan *green bean* menggunakan model *MobileNet* dengan arsitektur *Convolutional Neural Network* (CNN) berhasil diimplementasikan pada perangkat Android dan Website, serta bekerja baik dalam membedakan arabika, robusta, dan excelsa. Aplikasi ini membuktikan efektivitas *MobileNet* sebagai model yang ringan dan cepat untuk memproses gambar dan mengekstraksi fitur-fitur penting dari biji kopi. Hasil ini diharapkan dapat bermanfaat bagi industri kopi, khususnya bagi petani, pedagang, konsumen, dan *roaster* yang dapat mengidentifikasi jenis biji kopi secara cepat dan akurat. Penerapan teknologi ini membuka peluang luas untuk memanfaatkan *deep learning* dalam pengembangan aplikasi perkebunan yang dapat meningkatkan kualitas dan produktivitas secara berkelanjutan.

PUSTAKA

- Aini, N., & Liliana, D. Y. (2022). Prediksi Gender Berdasarkan Citra Mata Menggunakan Metode Convolutional Neural Network, Inception dan MobileNet. *Buletin Poltanesa*, 23(1). <https://doi.org/10.51967/tanesa.v23i1.1272>
- Anu, T. A., Rosnelly, R., Hasibuan, U., & Bulolo, P. (n.d.). KLASIFIKASI FITUR WARNA LEVEL ROASTING BIJI KOPI MENGGUNAKAN ARTIFICIAL NEURAL NETWORK. *JURNAL DEVICE*, 13(1), 8–13.
- Diantoni, C., Komarudin, O., Rizal Informatika, A., Karawang, S., Ronggo Waluyo, J. H., Timur, T., Karawang, J., & Barat, I. (2024). ARSITEKTUR MVVM DAN FRAMEWORK JETPACK COMPOSE PADA PENGEMBANGAN APLIKASI ANDROID (STUDI KASUS: APLIKASI SUKACOLAB). In *Jurnal Mahasiswa Teknik Informatika* (Vol. 8, Issue 3).
- Eingar, A. A., Arafa, M., Fathy, A., Moustafa, B., Mahmoud, O., Shaban, M., & Fawzy, N. (2021). Image Classification Based On CNN: A Survey. *Journal of Cybersecurity and Information Management (JCIM)*, 6(1), 18. <https://doi.org/10.5281/zenodo.4897990>
- Gumulya, D., & Helmi, I. S. (n.d.). KAJIAN BUDAYA MINUM KOPI INDONESIA.
- Hardi, N. (2022). Komparasi Algoritma MobileNet Dan Nasnet Mobile Pada Klasifikasi Penyakit Daun Teh. *Jurnal Rekayasa Perangkat Lunak*, 3(1).
- Hasan, I. (2022). ANALISIS SISTEM AGRIBISNIS KOPI ARABIKA DI DESA TOLAJUK, KECAMATAN LATIMOJONG, KABUPATEN LUWU. *WIRATANI: Jurnal Ilmiah Agribisnis*, 4(1), 2021. <http://jurnal.agribisnis.umi.ac.id>
- Khawas, C., & Shah, P. (2018). Application of Firebase in Android App Development-A Study. *International Journal of Computer Applications*, 179(46), 49–53. <https://doi.org/10.5120/ijca2018917200>

- Maulana, F., Afyenni, R., & Erianda, A. (2022). Aplikasi Manajemen Laboratorium Menggunakan Metode MVVM Berbasis Android. In *JITSI : Jurnal Ilmiah Teknologi Sistem Informasi* (Vol. 3, Issue 3). <http://jurnal-itsi.org>
- Michele, A., Colin, V., & Santika, D. D. (2019). Mobilenet convolutional neural networks and support vector machines for palmprint recognition. *Procedia Computer Science*, 157, 110–117. <https://doi.org/10.1016/j.procs.2019.08.147>
- Nugroho, M. A., & Sebatubun, M. M. (2020). KLASIFIKASI VARIETAS KOPI BERDASARKAN GREEN BEAN COFFEE MENGGUNAKAN METODE MACHINE LEARNING (Vol. 1, Issue 2).
- Riyadhi, I. M., Intan Purnamasari, & Kamal Prihandani. (2023). PENERAPAN POLA ARSITEKTUR MVVM PADA PERANCANGAN APLIKASI PENGADUAN MASYARAKAT BERBASIS ANDROID. *INFOTECH Journal*, 9(1), 147–158. <https://doi.org/10.31949/infotech.v9i1.5246>
- Royand, J., Simanullang, W., Zoromi2, F., Rio, U., & Nasution, T. (2024). APLIKASI PENJUALAN PRODUK IT DAN JASA SERVICE BERBASIS ANDROID DENGAN METODE MVVM. *Jurnal TEKINKOM*, 7(1). <https://doi.org/10.37600/tekinkom.v7i1.1241>
- Saputra, R. A., Wasyianti, S., Supriyatna, A., & Saefudin, D. F. (2021). Penerapan Algoritma Convolutional Neural Network Dan Arsitektur MobileNet Pada Aplikasi Deteksi Penyakit Daun Padi. *JURNAL SWABUMI*, 9(2). <https://archive.ics.uci.edu/ml/datasets/Rice>
- Schwermer, P. (2018). Performance Evaluation of Kotlin and Java on Android Runtime. In *DEGREE PROJECT COMPUTER SCIENCE AND ENGINEERING*.
- Sulistio, M. J., Lubis, C., & Kunci, K. (2023). Implementasi CNN dan MobileNet untuk Mendeteksi Penyakit Pneumonia dan COVID-19 dengan Menggunakan Aplikasi Smartphone. *Nusantara Journal of Multidisciplinary Science*, 1(4). <https://jurnal.intekom.id/index.php/njms>
- Sulkifly Said, M., Studi Sistem Komputer, P., Studi Sistem Informasi, P., & Catur sakti Kendari, S. (2024). KLASIFIKASI JENIS BERAS MENGGUNAKAN METODE CONVOLUTIONAL NEURAL NETWORK PADA ARSITEKTUR MOBILENET. 9(2).
- Suwitono, Y. A., & Kaunang, F. J. (2022). Implementasi Algoritma Convolutional Neural Network (CNN) Untuk Klasifikasi Daun Dengan Metode Data Mining SEMMA Menggunakan Keras. *Jurnal Komtika (Komputasi Dan Informatika)*, 6(2), 109–121. <https://doi.org/10.31603/komtika.v6i2.8054>
- Tayal, A., Gupta, J., Solanki, A., Bisht, K., Nayyar, A., & Masud, M. (2022). DL-CNN-based approach with image processing techniques for diagnosis of retinal diseases. *Multimedia Systems*, 28(4), 1417–1438. <https://doi.org/10.1007/s00530-021-00769-7>
- Thoma, M. (2017). *Analysis and Optimization of Convolutional Neural Network Architectures*. <http://arxiv.org/abs/1707.09725>
- Zhu, N., Liu, X., Liu, Z., Hu, K., Wang, Y., Tan, J., Huang, M., Zhu, Q., Ji, X., Jiang, Y., & Guo, Y. (2018). Deep learning for smart agriculture: Concepts, tools, applications, and opportunities. *International Journal of Agricultural and Biological Engineering*, 11(4), 21–28. <https://doi.org/10.25165/j.ijabe.20181104.4475>