



Implementasi MobileNetV2 Pada Aplikasi Forensik Android Untuk Deteksi Citra AI-generated dengan Ketahanan Terhadap Transformasi Citra

Aryanahta Putra*, Resmi Darni, Dony Novaliendry, Vikri Aulia

Fakultas Teknik, Informatika, Univesitas Negeri Padang, Padang

Jl. Prof. Dr. Hamka, Air Tawar Bar., Kec. Padang Utara, Kota Padang, Sumatera Barat, Indonesia

Email: ^{1,*}arya272018@gmail.com, ²resmidarni@ft.unp.ac.id, ³dony.novaliendry@ft.unp.ac.id, ⁴vikriaulia@unp.ac.id

Email Penulis Korespondensi: arya272018@gmail.com

Submitted: 08/07/2026; Accepted: 25/07/2026; Published: 27/07/2026

Abstrak—Perkembangan Generative Artificial Intelligence menghasilkan citra sintetis yang semakin realistis sehingga sulit dibedakan dari citra otentik melalui pengamatan visual. Penelitian ini bertujuan mengimplementasikan aplikasi mobile forensik digital berbasis Android untuk deteksi awal citra AI-generate d menggunakan MobileNetV2 yang dikonversi ke TensorFlow Lite agar inferensi dapat berjalan secara on-device. Penelitian menggunakan pendekatan kuantitatif-eksperimental dengan 2.000 citra, terdiri atas 1.000 citra otentik dan 1.000 citra AI-generate d. Citra otentik berasal dari 500 foto kamera smartphone dan 500 citra GenImage, sedangkan citra AI-generate d berasal dari Stable Diffusion v1.4, Stable Diffusion v1.5, Wukong, dan Midjourney. Dataset dibagi menjadi 1.000 data latih, 500 data validasi, dan 500 data uji. Data latih digunakan untuk pelatihan, data validasi untuk pemilihan model dan penentuan threshold, sedangkan data uji digunakan hanya untuk evaluasi akhir. Pengujian robustness dilakukan pada salinan 500 data uji tanpa pelatihan ulang. Pada kondisi normal dengan threshold 0,680, model memperoleh accuracy 85,40%, precision 85,53%, recall 85,40%, dan F1-score 85,39%. Kompresi JPEG q=65 menghasilkan accuracy 86,00%. Kombinasi resize 112 × 112 piksel dan JPEG q=65 menghasilkan penurunan terbesar, dengan accuracy 62,40% atau turun 23,00 poin persentase dan F1-score 56,66%. Aplikasi berhasil menjalankan inferensi lokal, menampilkan label dan confidence score, serta menyimpan riwayat deteksi. Kontribusi penelitian ini adalah integrasi deteksi on-device dan evaluasi robustness menggunakan subset uji yang konsisten, sehingga aplikasi dapat digunakan sebagai alat bantu deteksi awal, bukan sebagai verifikasi forensik final.

Kata Kunci: Aplikasi Mobile; Citra AI-Generate D; Forensik Digital; MobileNetV2; Robustness Testing

Abstract—The development of Generative Artificial Intelligence has produced realistic synthetic images that are difficult to distinguish from authentic images through visual inspection. This study aims to implement an Android-based mobile digital forensics application for detecting AI-generate d images using MobileNetV2 converted to TensorFlow Lite for on-device inference. A quantitative-experimental approach used 2,000 images, consisting of 1,000 authentic and 1,000 AI-generate d images. The authentic class comprised 500 smartphone photographs and 500 GenImage samples, while the AI-generate d class included Stable Diffusion v1.4, Stable Diffusion v1.5, Wukong, and Midjourney images. The dataset was divided into 1,000 training, 500 validation, and 500 testing images. Training data were used for model development, validation data for model selection and threshold determination, and testing data only for final evaluation. Robustness testing used copies of 500 testing images without retraining. Under normal conditions with a threshold of 0.680, the model achieved 85.40% accuracy, 85.53% precision, 85.40% recall, and an 85.39% F1-score. JPEG q=65 compression produced 86.00% accuracy. The largest degradation occurred with 112 × 112 resizing combined with JPEG q=65, resulting in 62.40% accuracy, a decrease of 23.00 percentage points, and a 56.66% F1-score. The application performed local inference, displayed prediction labels and confidence scores, and stored detection history. This study contributes an on-device detection application and robustness evaluation using a consistent test subset, positioning the system as an initial detection aid rather than a final forensic verification tool.

Keywords: AI-Generate D Image; Digital Forensics; Mobile Application; MobileNetV2; Robustness Testing

1. PENDAHULUAN

Perkembangan Generative Artificial Intelligence telah meningkatkan kemampuan sistem komputasional dalam menghasilkan citra sintetis dengan kualitas visual yang realistis. Model generatif modern, khususnya diffusion model, mampu menghasilkan citra beresolusi tinggi dengan detail, tekstur, pencahayaan, dan komposisi yang menyerupai hasil perekaman dunia nyata [1]. GenImage sebagai benchmark juga menunjukkan bahwa citra yang dihasilkan oleh Generative Adversarial Network maupun diffusion model memiliki keragaman konten dan kualitas visual yang semakin meyakinkan [2]. Perkembangan tersebut bermanfaat bagi desain, pendidikan, hiburan, pemasaran, dan industri kreatif. Namun, kemampuan menghasilkan citra sintetis yang semakin realistis turut menimbulkan tantangan terhadap verifikasi keaslian informasi visual karena asal pembentukan suatu gambar tidak selalu dapat dikenali melalui pengamatan langsung.

Citra digital dapat tersebar dengan cepat melalui media sosial, aplikasi pesan, situs berita, dan platform daring. Selama proses distribusi, gambar dapat disalin, disunting, dikompresi, diubah ukurannya, atau diunggah kembali tanpa informasi sumber yang memadai. Citra yang tampak realistis belum tentu berasal dari perekaman objek dunia nyata menggunakan kamera, tetapi dapat dibentuk melalui proses komputasional. Penggunaan citra sintetis tanpa keterangan yang jelas berpotensi menimbulkan kesalahan persepsi, misinformasi, manipulasi opini, dan penurunan kepercayaan terhadap konten digital. Kondisi tersebut menunjukkan perlunya alat bantu yang dapat



melakukan pemeriksaan awal terhadap citra otentik dan citra AI-generate d secara cepat, praktis, serta mudah diakses.

Kemampuan manusia dalam membedakan citra otentik dan citra sintetis juga tidak selalu konsisten. Nightingale dan Farid menunjukkan bahwa pada kondisi tertentu, wajah sintetis dapat dinilai sama realistiknya atau bahkan lebih meyakinkan daripada wajah nyata oleh pengamat manusia [3]. Temuan tersebut memperlihatkan bahwa pemeriksaan visual tidak cukup digunakan sebagai satu-satunya dasar untuk menilai asal pembentukan citra. Dalam konteks forensik digital, gambar dipahami sebagai artefak yang memiliki pola, karakteristik, dan jejak pembentukan tertentu. Oleh karena itu, diperlukan pendekatan komputasional yang mampu mempelajari perbedaan visual secara sistematis. Deep learning menjadi salah satu pendekatan yang banyak digunakan karena mampu membentuk representasi fitur secara otomatis dari data pelatihan [4], [5].

Sejumlah penelitian telah mengembangkan detektor citra AI-generate d dengan pendekatan yang berbeda. Ojha et al. mengembangkan detektor yang diarahkan untuk meningkatkan generalisasi terhadap keluaran generator yang tidak digunakan selama pelatihan [6]. Corvi et al. menganalisis citra sintetis hasil diffusion model dan menunjukkan bahwa perubahan kualitas gambar dapat memengaruhi kemampuan detektor [7]. Wang et al. mengusulkan DIRE dengan memanfaatkan perbedaan rekonstruksi antara citra nyata dan citra hasil diffusion model [8]. Cozzolino et al. menggunakan representasi CLIP untuk meningkatkan generalisasi terhadap generator yang berbeda [9]. Pendekatan lain memanfaatkan karakteristik up-sampling dan kemiripan spektrum citra untuk meningkatkan kemampuan generalisasi model [10], [11]. AEROBLADE menggunakan galat rekonstruksi autoencoder untuk mendeteksi citra latent diffusion tanpa melatih detektor tambahan [12]. Penelitian berikutnya menerapkan Variational Information Bottleneck dan strategi pelatihan bebas bias untuk mengurangi informasi yang tidak relevan serta meningkatkan generalisasi model [13], [14].

Aspek ketahanan model juga penting karena citra pada kondisi penggunaan nyata dapat mengalami kompresi, perubahan resolusi, atau pemrosesan ulang. Bammey menunjukkan bahwa kompresi JPEG dan perbedaan sumber generator perlu dipertimbangkan ketika mengevaluasi detektor citra hasil diffusion model [15]. Li et al. menyoroti adanya kesenjangan antara performa detektor pada lingkungan pengujian ideal dan kondisi penggunaan nyata yang kompleks [16]. Perubahan resolusi dan keragaman generator juga dapat memengaruhi kemampuan generalisasi model [17], [18]. Selain itu, distribusi warna dan karakteristik frekuensi dapat dimanfaatkan untuk memperkuat deteksi citra sintetis [19], [20]. Pendekatan yang menggabungkan informasi tekstur dan konten juga digunakan untuk meningkatkan keandalan klasifikasi citra AI-generate d [21]. Temuan tersebut menunjukkan bahwa evaluasi pada citra normal belum cukup untuk menggambarkan ketahanan detektor secara menyeluruh.

Meskipun penelitian terdahulu telah menghasilkan berbagai metode deteksi yang efektif, sebagian besar masih berfokus pada pengembangan algoritma, generalisasi lintas generator, atau evaluasi model di lingkungan komputasi penelitian [6], [9]. Beberapa metode memanfaatkan proses rekonstruksi, representasi spektral, atau model visual berukuran relatif besar sehingga penerapannya pada perangkat dengan sumber daya terbatas memerlukan pertimbangan tambahan [8], [12]. Implementasi detektor dalam bentuk aplikasi Android yang dapat digunakan langsung oleh pengguna juga belum menjadi fokus utama penelitian tersebut. Selain itu, evaluasi yang menggabungkan kompresi JPEG, perubahan ukuran, kecerahan, kontras, rotasi, dan transformasi gabungan menggunakan identitas data uji yang sama masih perlu dikembangkan. Penggunaan subset uji yang berbeda pada setiap skenario dapat menyulitkan perbandingan karena perubahan hasil tidak hanya dipengaruhi transformasi, tetapi juga perbedaan isi data.

Penelitian ini menawarkan tiga perbedaan utama. Pertama, model tidak hanya dievaluasi di lingkungan eksperimen, tetapi diintegrasikan ke dalam aplikasi mobile forensik digital berbasis Android. Kedua, inferensi dijalankan secara lokal menggunakan TensorFlow Lite sehingga citra tidak perlu dikirim ke server eksternal. Ketiga, pengujian robustness menggunakan salinan dari subset data uji yang sama pada seluruh skenario tanpa mengubah bobot model, label kelas, dan threshold. Rancangan tersebut memungkinkan perubahan metrik dikaitkan secara langsung dengan pengaruh transformasi citra. Dengan demikian, penelitian tidak hanya menilai kemampuan klasifikasi model pada kondisi normal, tetapi juga mengamati keterbatasannya ketika kualitas input berubah selama penyimpanan dan distribusi digital.

Untuk memenuhi kebutuhan tersebut, penelitian menggunakan MobileNetV2 sebagai arsitektur klasifikasi biner antara citra otentik dan citra AI-generate d. MobileNetV2 dirancang untuk perangkat dengan sumber daya terbatas melalui penggunaan depthwise separable convolution, inverted residuals, dan linear bottlenecks [22]. Arsitektur ini dipilih karena menawarkan keseimbangan antara kemampuan ekstraksi fitur visual dan efisiensi komputasi. Model diterapkan melalui transfer learning dengan memanfaatkan bobot prelatih ImageNet, kemudian dikonversi ke format TensorFlow Lite agar inferensi dapat dijalankan secara on-device pada perangkat Android [23]. Pendekatan tersebut mengurangi ketergantungan terhadap koneksi internet sekaligus mendukung pemrosesan citra lokal.

Aplikasi yang dikembangkan mengintegrasikan pemilihan citra dari galeri atau kamera, validasi masukan, pra-pemrosesan, inferensi model, penampilan label prediksi dan confidence score, serta penyimpanan riwayat deteksi secara lokal. Dataset penelitian berjumlah 2.000 citra dengan komposisi seimbang, terdiri atas 1.000 citra otentik dan 1.000 citra AI-generate d. Citra otentik berasal dari kamera smartphone dan subset otentik GenImage, sedangkan citra AI-generate d berasal dari Stable Diffusion v1.4, Stable Diffusion v1.5, Wukong, dan Midjourney

[2]. Dataset dibagi menjadi data latih, data validasi, dan data uji agar proses pelatihan, pemilihan model, penentuan threshold, dan evaluasi akhir dilakukan secara terpisah.

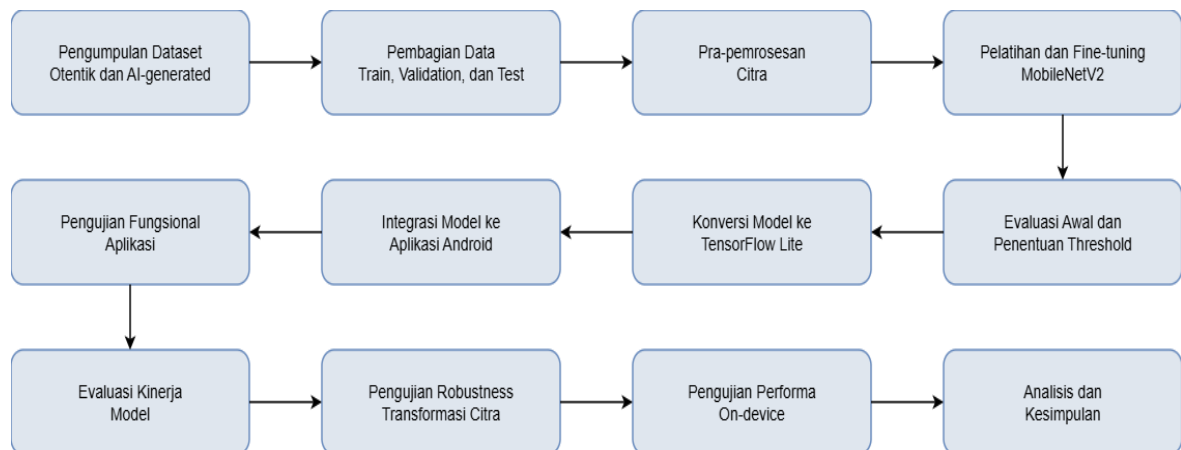
Evaluasi dilakukan pada kondisi normal dan berbagai skenario robustness yang mencakup kompresi JPEG, perubahan resolusi, perubahan brightness, perubahan contrast, rotasi, serta transformasi gabungan. Setiap skenario menggunakan salinan dari 500 citra pada subset data uji yang sama tanpa pelatihan ulang. Kinerja model dianalisis menggunakan accuracy, precision, recall, F1-score, confusion matrix, False Positive, dan False Negative. Selain itu, pengujian performa on-device dilakukan pada dua perangkat Android dengan spesifikasi berbeda untuk menilai waktu inferensi dan kestabilan aplikasi.

Kontribusi penelitian ini terletak pada integrasi MobileNetV2 ke dalam aplikasi Android yang mampu melakukan deteksi awal citra AI-generate d secara on-device serta evaluasi robustness menggunakan subset data uji yang konsisten. Penggunaan data uji yang sama memungkinkan hasil antarskenario dibandingkan secara lebih objektif, sedangkan penyediaan label prediksi, confidence score, dan riwayat lokal membuat keluaran model lebih mudah diakses oleh pengguna. Berdasarkan uraian tersebut, penelitian ini bertujuan mengimplementasikan MobileNetV2 pada aplikasi mobile forensik digital berbasis Android, mengevaluasi kinerja klasifikasi dan ketahanannya terhadap transformasi citra, serta menilai kelayakan penerapannya sebagai alat bantu deteksi awal. Aplikasi tidak dimaksudkan sebagai alat verifikasi forensik final atau satu-satunya dasar pengambilan keputusan.

2. METODOLOGI PENELITIAN

2.1 Tahapan Penelitian

Penelitian ini menggunakan pendekatan kuantitatif-eksperimental untuk mengembangkan dan mengevaluasi aplikasi deteksi citra AI-generate d berbasis Android. Kinerja model diukur melalui metrik klasifikasi, pengujian robustness, dan pengujian inferensi pada perangkat Android. Tahapan penelitian dimulai dari pengumpulan dataset hingga analisis hasil, sebagaimana ditunjukkan pada Gambar 1.



Gambar 1. Tahapan Penelitian

Berdasarkan Gambar 1, penelitian dilaksanakan melalui tahapan berikut:

1. **Pengumpulan dataset**
Dataset terdiri atas citra otentik dan citra AI-generate d. Citra otentik diperoleh dari kamera smartphone dan subset otentik GenImage, sedangkan citra AI-generate d berasal dari Stable Diffusion v1.4, Stable Diffusion v1.5, Wukong, dan Midjourney [2]. Seluruh citra diperiksa untuk memastikan berkas dapat dibaca, label sesuai, dan tidak terdapat duplikasi identik.
2. **Pembagian data**
Sebanyak 2.000 citra dibagi menjadi 1.000 data train, 500 data validation, dan 500 data test dengan komposisi kelas yang seimbang. Data train digunakan untuk pelatihan, data validation digunakan untuk memilih model dan menentukan threshold, sedangkan data test hanya digunakan untuk evaluasi akhir. Pemisahan subset dilakukan untuk mencegah data leakage.
3. **Pra-pemrosesan citra**
Setiap citra dikonversi ke format RGB, diubah menjadi ukuran 224×224 piksel, diproses menggunakan fungsi pra-pemrosesan MobileNetV2, dan dibentuk menjadi tensor masukan. Tahapan yang sama diterapkan pada proses pelatihan, evaluasi, dan inferensi dalam aplikasi.
4. **Pelatihan MobileNetV2**
MobileNetV2 digunakan sebagai backbone dengan pendekatan transfer learning. Bobot pralatih ImageNet dimanfaatkan untuk mengekstraksi fitur visual, kemudian kepala klasifikasi disesuaikan untuk membedakan citra otentik dan AI-generate d. MobileNetV2 menggunakan inverted residuals dan linear bottlenecks yang mendukung efisiensi komputasi pada perangkat mobile [22].

5. Evaluasi awal model
Performa model dipantau menggunakan data validation. Model terbaik dipilih berdasarkan hasil validasi, kemudian probabilitas prediksi pada subset tersebut digunakan untuk menentukan threshold klasifikasi. Data test tidak digunakan dalam pemilihan model maupun penentuan threshold.
6. Konversi ke TensorFlow Lite
Model terbaik dikonversi dari format TensorFlow/Keras menjadi berkas .tflite. Bentuk input, tipe data, dan keluaran probabilitas diperiksa untuk memastikan model hasil konversi tetap sesuai dengan rancangan klasifikasi. TensorFlow Lite digunakan agar inferensi dapat dijalankan secara lokal pada perangkat Android [23].
7. Integrasi aplikasi Android
Model TensorFlow Lite diintegrasikan ke aplikasi Flutter. Aplikasi menerima citra dari galeri atau kamera, melakukan pra-pemrosesan, menjalankan inferensi secara on-device, menampilkan label dan confidence score, serta menyimpan riwayat deteksi secara lokal.
8. Pengujian fungsional
Pengujian fungsional dilakukan menggunakan metode black box testing. Pengujian mencakup pemilihan citra, pengambilan gambar melalui kamera, proses deteksi, penampilan hasil, penyimpanan riwayat, dan penghapusan riwayat. Skenario negatif juga diuji, seperti input kosong, berkas tidak valid, penolakan izin akses, dan kondisi riwayat kosong.
9. Evaluasi kinerja model
Evaluasi akhir menggunakan 500 citra test yang terdiri atas 250 citra otentik dan 250 citra AI-generate d. Citra asli diuji sebagai kondisi normal atau baseline. Kinerja model dianalisis menggunakan accuracy, precision, recall, F1-score, confusion matrix, False Positive, dan False Negative.
10. Pengujian robustness
Pengujian robustness menggunakan salinan dari 500 citra test yang sama tanpa pelatihan ulang. Skenario pengujian mencakup JPEG q=65 dan q=30, resize 112×112 dan 64×64 , kombinasi resize 112×112 dengan JPEG q=65, brightness 0,7 dan 1,3, contrast 0,7 dan 1,3, serta rotation -10° dan $+10^\circ$. Perubahan kualitas dan kompresi citra dapat memengaruhi kemampuan detektor dalam mengenali citra sintetis [7], [15].
Pada seluruh skenario, label kelas, bobot model, dan threshold tetap sama. Dengan demikian, perubahan hasil pengujian mencerminkan pengaruh transformasi citra dan bukan akibat perbedaan data atau pelatihan ulang.
11. Pengujian performa on-device
Aplikasi diuji pada dua perangkat Android dengan spesifikasi berbeda. Pengujian mencakup keberhasilan inferensi lokal, waktu inferensi, penampilan hasil prediksi, dan kestabilan aplikasi.
12. Analisis hasil
Hasil pengujian fungsional, evaluasi kondisi normal, pengujian robustness, dan pengujian on-device dianalisis untuk mengetahui kemampuan dan keterbatasan sistem. Hasil tersebut digunakan untuk menilai kelayakan aplikasi sebagai alat bantu deteksi awal, bukan sebagai alat verifikasi forensik final.

2.2 Penerapan MobileNetV2 dan TensorFlow Lite

MobileNetV2 digunakan sebagai model utama untuk melakukan klasifikasi biner antara citra otentik dan citra AI-generate d. Arsitektur ini dirancang untuk perangkat dengan sumber daya komputasi terbatas melalui penggunaan depthwise separable convolution, inverted residuals, dan linear bottlenecks [22]. Model diterapkan menggunakan pendekatan transfer learning dengan memanfaatkan bobot pralatih ImageNet. MobileNetV2 berperan sebagai backbone untuk mengekstraksi fitur visual, sedangkan kepala klasifikasinya terdiri atas Global Average Pooling, Batch Normalization, Dropout 0,4, Dense 128 dengan aktivasi ReLU dan regularisasi L2, Batch Normalization, Dropout 0,3, serta satu neuron keluaran dengan aktivasi sigmoid. Nilai keluaran sigmoid menunjukkan probabilitas citra termasuk kelas AI-generate d.

Model menerima citra RGB berukuran 224×224 piksel. Nilai piksel diproses menggunakan fungsi MobileNetV2 preprocess_input sehingga berada pada rentang -1 sampai 1 . Pelatihan dilakukan dalam dua tahap menggunakan batch size 16. Tahap pertama melatih kepala klasifikasi selama 15 epoch dengan learning rate 0,0001. Tahap kedua melakukan fine-tuning terhadap 25 lapisan terakhir MobileNetV2 selama 12 epoch dengan learning rate 0,000005. Data train digunakan untuk pelatihan, sedangkan data validation digunakan untuk memilih model berdasarkan weighted F1-score dan menentukan threshold. Data test tidak digunakan dalam pemilihan model maupun penentuan threshold. Model terpilih kemudian dikonversi dari format TensorFlow/Keras menjadi TensorFlow Lite berformat float32. Model TFLite menerima tensor berukuran $1 \times 224 \times 224 \times 3$ dan menghasilkan satu nilai probabilitas kelas AI-generate d. Threshold sebesar 0,680 digunakan sebagai batas keputusan, yaitu citra diklasifikasikan sebagai AI-generate d apabila probabilitasnya lebih besar atau sama dengan 0,680 dan sebagai citra otentik apabila probabilitasnya lebih rendah. TensorFlow Lite digunakan agar inferensi dapat dijalankan secara lokal pada perangkat Android [23]. Berkas .tflite selanjutnya digunakan dalam aplikasi Flutter untuk memproses citra dari galeri atau kamera, menampilkan label prediksi dan confidence score, serta mendukung penyimpanan riwayat deteksi lokal.

3. HASIL DAN PEMBAHASAN

3.1 Dataset dan Pembagian Data

Dataset penelitian digunakan untuk membentuk klasifikasi biner antara citra otentik dan citra AI-generate d. Dataset berjumlah 2.000 citra dengan komposisi seimbang, yaitu 1.000 citra otentik dan 1.000 citra AI-generate d. Keseimbangan kedua kelas diterapkan untuk mengurangi kecenderungan model memprediksi salah satu kelas secara berlebihan. Citra otentik berasal dari 500 foto hasil akuisisi menggunakan kamera smartphone dan 500 citra dari subset otentik GenImage. Sementara itu, citra AI-generate d diperoleh dari Stable Diffusion v1.4, Stable Diffusion v1.5, Wukong, dan Midjourney. GenImage menyediakan citra otentik dan sintetis dari beberapa model generatif yang dapat digunakan dalam penelitian deteksi citra AI-generate d [2]. Rincian sumber dan komposisi dataset disajikan pada Tabel 1.

Tabel 1. Sumber dan Komposisi Dataset Penelitian

Kelas	Sumber/Generator	Jumlah	Proporsi Dalam Kelas	Proporsi Total
Otentik	Kamera Smartphone	500	50%	25%
Otentik	GenImage	500	50%	25%
Subtotal Otentik		1.000	100%	50%
AI-generate d	Stable Diffusion v1.4	300	30%	15%
AI-generate d	Stable Diffusion v1.5	300	30%	15%
AI-generate d	Wukong	300	30%	15%
AI-generate d	Midjourney	100	10%	5%
Subtotal AI-generate d		1.000	100%	50%
Total Dataset		2.000		100%

Berdasarkan Tabel 1, kelas otentik dan kelas AI-generate d masing-masing memiliki proporsi 50% dari keseluruhan dataset. Kelas otentik menggabungkan citra hasil kamera smartphone dan GenImage, sedangkan kelas AI-generate d berasal dari empat generator. Penggunaan beberapa sumber bertujuan meningkatkan keragaman karakteristik citra dan mengurangi ketergantungan model terhadap satu sumber atau generator tertentu.

Sebelum pembagian data, seluruh citra diperiksa untuk memastikan berkas dapat dibaca, label kelas sesuai, dan tidak terdapat duplikasi identik. Dataset kemudian dipisahkan menjadi data latih, data validasi, dan data uji. Rincian pembagian dataset ditunjukkan pada Tabel 2.

Tabel 2. Pembagian Dataset Penelitian

Subset	Citra Otentik	Citra AI-generate d	Total	Proporsi
Data latih	500	500	1.000	50%
Data validasi	250	250	500	25%
Data uji	250	250	500	25%
Total	1.000	1.000	2.000	100%

Berdasarkan Tabel 2, setiap subset memiliki jumlah kedua kelas yang seimbang. Data latih digunakan untuk melatih dan melakukan fine-tuning model MobileNetV2. Data validasi digunakan untuk memantau proses pelatihan, memilih model terbaik, dan menentukan threshold klasifikasi. Data uji tidak digunakan dalam pelatihan, pemilihan model, maupun penentuan threshold, tetapi hanya digunakan setelah konfigurasi model ditetapkan agar evaluasi akhir tetap objektif. Pemisahan antarsubset juga memastikan bahwa citra pada data latih tidak muncul kembali dalam data validasi maupun data uji sehingga risiko kebocoran data dapat dikurangi.

Sebanyak 500 citra pada data uji digunakan dalam evaluasi kondisi normal sebagai baseline. Pengujian robustness selanjutnya menggunakan salinan dari 500 citra uji yang sama. Pada setiap skenario, salinan citra diberikan transformasi berupa kompresi JPEG, resize, kombinasi resize dan JPEG, perubahan brightness, perubahan contrast, atau rotation. Transformasi hanya diterapkan pada input pengujian tanpa mengubah citra asli dan label kelas.

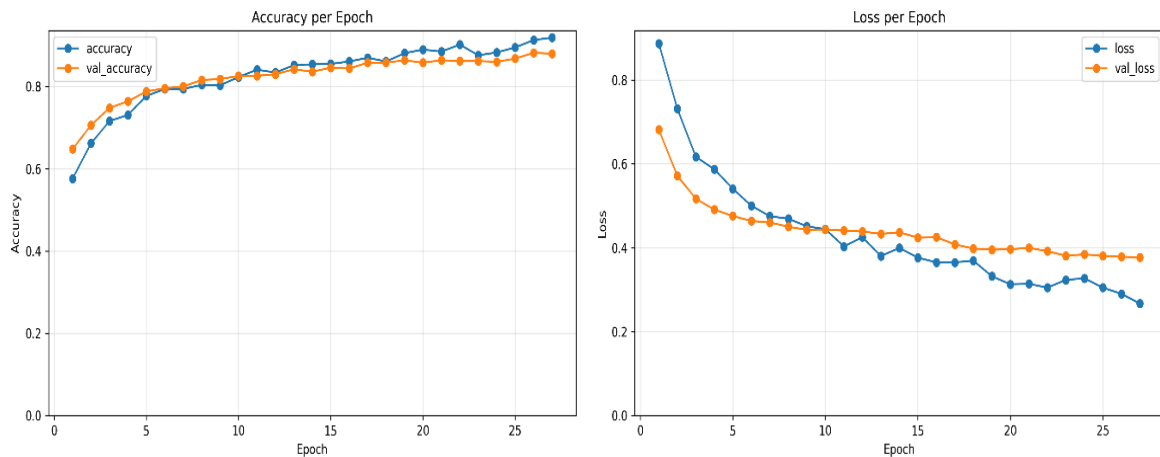
Seluruh skenario robustness menggunakan bobot model dan threshold yang sama serta tidak melibatkan pelatihan ulang. Dengan demikian, data robustness bukan bagian dari data latih dan bukan kumpulan data baru, melainkan versi tertransformasi dari data uji yang telah dipisahkan sebelum proses pelatihan. Penggunaan identitas citra yang sama memungkinkan perubahan performa antarskenario dikaitkan dengan pengaruh transformasi, bukan dengan perbedaan komposisi data.

3.2 Hasil Pelatihan dan Konversi Model MobileNetV2

Model MobileNetV2 dilatih untuk melakukan klasifikasi biner antara citra otentik dan citra AI-generate d. Data train digunakan untuk pelatihan model, sedangkan data validation digunakan untuk memantau performa, memilih model terbaik, dan menentukan threshold klasifikasi. Sebelum diproses, citra dikonversi ke format RGB,

diubah menjadi ukuran 224×224 piksel, dan diproses menggunakan fungsi MobileNetV2 preprocess_input sehingga nilai piksel berada pada rentang -1 sampai 1 .

Pelatihan dilakukan dalam dua tahap menggunakan batch size 16. Tahap pertama merupakan pelatihan kepala klasifikasi selama 15 epoch dengan learning rate 0,0001, sementara lapisan backbone MobileNetV2 dibekukan. Tahap kedua berupa fine-tuning terhadap 25 lapisan terakhir selama 12 epoch dengan learning rate 0,000005. Augmentasi hanya diterapkan pada data train, sedangkan data validation tidak diberi augmentasi. Perubahan nilai accuracy dan loss selama kedua tahap pelatihan disajikan pada Gambar 2.



Gambar 2. Grafik Accuracy dan Loss Pelatihan Dua Tahap MobileNetV2

Berdasarkan Gambar 2, pada tahap pelatihan kepala klasifikasi, training accuracy meningkat dari 57,60% pada epoch pertama menjadi 85,50% pada epoch ke-15. Pada tahap yang sama, validation accuracy meningkat dari 64,80% menjadi 84,60%. Nilai training loss juga menurun dari 0,887 menjadi 0,376, sedangkan validation loss menurun dari 0,682 menjadi 0,424. Hasil tersebut menunjukkan bahwa kepala klasifikasi mampu mempelajari perbedaan antara citra otentik dan citra AI-generate d secara bertahap.

Proses fine-tuning selanjutnya meningkatkan kemampuan model dengan menyesuaikan sebagian fitur pada lapisan akhir MobileNetV2. Pada akhir fine-tuning, training accuracy mencapai 91,90% dan validation accuracy pada batas keputusan standar 0,5 mencapai 88,00%. Nilai training loss menurun menjadi 0,267, sedangkan validation loss menjadi 0,377. Walaupun performa data train lebih tinggi daripada data validation, performa validasi tetap meningkat hingga tahap akhir sehingga model terpilih berasal dari fine-tuning epoch ke-12.

Pemilihan model dilakukan berdasarkan nilai weighted F1-score pada data validation. Probabilitas prediksi pada subset tersebut digunakan untuk menentukan threshold terbaik sebesar 0,680. Dengan threshold tersebut, model memperoleh validation accuracy sebesar 88,40%, precision weighted 88,40%, recall weighted 88,40%, dan F1-score weighted 88,40%. Nilai ROC-AUC validasi mencapai 93,79%, sedangkan average precision mencapai 93,14%. Data test tidak digunakan dalam pemilihan model maupun penentuan threshold, sehingga tetap digunakan secara terpisah untuk evaluasi akhir.

Model terpilih kemudian dikonversi dari format TensorFlow/Keras menjadi TensorFlow Lite berformat float32. Model hasil konversi memiliki ukuran sekitar 9,07 MB, menerima tensor masukan berukuran $1 \times 224 \times 224 \times 3$, dan menghasilkan satu nilai probabilitas kelas AI-generate d. Citra diklasifikasikan sebagai AI-generate d apabila nilai probabilitas lebih besar atau sama dengan 0,680, sedangkan nilai di bawah batas tersebut diklasifikasikan sebagai citra otentik.

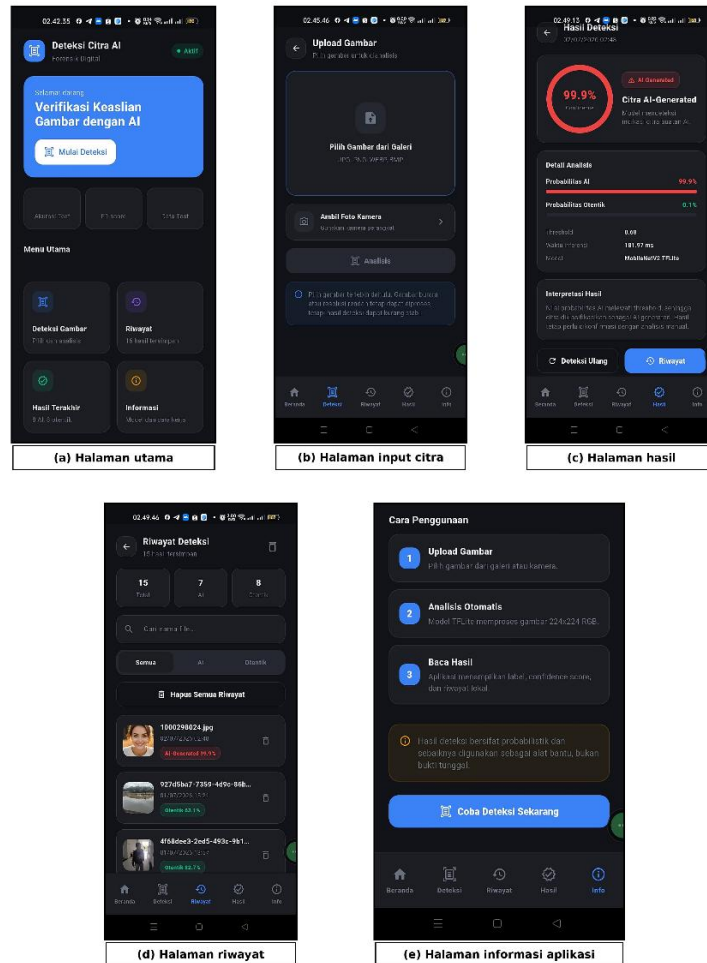
Kesesuaian model TensorFlow Lite diperiksa menggunakan seluruh 500 citra pada data test. Hasil pemeriksaan menunjukkan tidak terdapat perbedaan keputusan klasifikasi antara model TensorFlow/Keras dan TensorFlow Lite. Selisih probabilitas maksimum hanya sebesar 0,00000369 dan seluruh metrik evaluasi yang dibandingkan menghasilkan nilai yang sama. Hasil tersebut menunjukkan bahwa proses konversi berhasil mempertahankan keluaran klasifikasi model sehingga model TensorFlow Lite layak digunakan pada tahap integrasi aplikasi Android.

3.3 Hasil implementasi Aplikasi

Hasil implementasi penelitian berupa aplikasi mobile berbasis Android yang digunakan sebagai alat bantu deteksi awal citra otentik dan citra AI-generate d. Aplikasi dikembangkan menggunakan Flutter dan Dart dengan mengintegrasikan model MobileNetV2 berformat TensorFlow Lite. Proses inferensi dijalankan secara on-device sehingga citra dapat dianalisis langsung pada perangkat tanpa dikirim ke server eksternal.

Alur penggunaan dimulai ketika pengguna memilih citra melalui galeri atau mengambil gambar menggunakan kamera. Citra kemudian dikonversi ke format RGB, diubah menjadi ukuran 224×224 piksel, dan diproses menggunakan pra-pemrosesan MobileNetV2 sebelum dibentuk menjadi tensor masukan. Model

TensorFlow Lite menghasilkan nilai probabilitas kelas AI-generate d. Berdasarkan threshold 0,680, citra diklasifikasikan sebagai AI-generate d apabila probabilitasnya lebih besar atau sama dengan batas tersebut, sedangkan nilai yang lebih rendah diklasifikasikan sebagai citra otentik. Hasil inferensi ditampilkan dalam bentuk label prediksi dan confidence score, kemudian disimpan secara lokal pada halaman riwayat. Tampilan hasil implementasi aplikasi disajikan pada Gambar 3.



Gambar 3. Tampilan Aplikasi Deteksi Citra AI-generate d

Berdasarkan Gambar 3, aplikasi terdiri atas lima tampilan utama. Gambar 3(a) menunjukkan halaman utama yang menyediakan akses menuju fitur deteksi, riwayat, hasil terakhir, dan informasi aplikasi. Gambar 3(b) merupakan halaman input citra yang digunakan untuk memilih gambar dari galeri atau mengambil foto melalui kamera. Gambar 3(c) menampilkan hasil deteksi berupa label prediksi, probabilitas setiap kelas, confidence score, dan informasi proses inferensi. Nilai persentase pada halaman tersebut merupakan tingkat keyakinan model terhadap satu citra yang diperiksa, bukan nilai akurasi keseluruhan model.

Gambar 3(d) memperlihatkan halaman riwayat yang menyimpan hasil pemeriksaan sebelumnya secara lokal. Pengguna dapat melihat kembali citra, label prediksi, dan waktu pemeriksaan serta menghapus riwayat yang tidak diperlukan. Gambar 3(e) merupakan halaman informasi aplikasi yang menjelaskan cara penggunaan, alur analisis, dan batasan hasil deteksi. Kelima tampilan tersebut menunjukkan bahwa fungsi pemilihan citra, inferensi lokal, penampilan hasil, dan penyimpanan riwayat telah dihubungkan dalam satu alur penggunaan aplikasi Android.

3.4 Hasil Pengujian Fungsional Aplikasi

Pengujian fungsional dilakukan menggunakan metode black-box testing untuk memastikan bahwa setiap fungsi aplikasi menghasilkan keluaran yang sesuai dengan masukan dan rancangan sistem. Pengujian mencakup 11 skenario fungsi utama dan 10 skenario negatif atau kondisi nonideal. Skenario fungsi utama meliputi navigasi aplikasi, pemilihan citra, pelaksanaan inferensi, penampilan hasil deteksi, pengelolaan riwayat, dan penyajian informasi aplikasi. Sementara itu, skenario negatif digunakan untuk mengamati respons aplikasi terhadap input kosong, berkas yang tidak dapat diproses, penolakan izin akses, model yang belum siap, dan kondisi riwayat kosong. Ringkasan hasil pengujian fungsional disajikan pada Tabel 3.

Tabel 3. Ringkasan Hasil Pengujian Fungsional Aplikasi

No.	Kelompok Pengujian	Jumlah Skenario	Cakupan Pengujian	Hasil
1	Navigasi dan informasi	3	Membuka aplikasi, menampilkan halaman informasi, dan menampilkan batasan penggunaan	Berhasil
2	Input citra	2	Memilih citra melalui galeri dan mengambil citra menggunakan kamera	Berhasil
3	Deteksi dan keluaran model	3	Menjalankan inferensi TensorFlow Lite, menampilkan label prediksi, dan menampilkan confidence score	Berhasil
4	Pengelolaan riwayat	3	Menyimpan, menampilkan, dan menghapus riwayat deteksi	Berhasil
5	Input tidak valid atau nonideal	5	Tanpa citra, berkas tidak dapat diproses, citra buram, resolusi rendah, dan ukuran berkas besar	Berhasil ditangani
6	Izin dan kesiapan sistem	3	Akses kamera ditolak, akses media tidak tersedia, dan model belum siap	Berhasil ditangani
7	Kondisi khusus riwayat	2	Riwayat kosong dan penghapusan seluruh riwayat	Berhasil ditangani
	Total	21	11 fungsi utama dan 10 skenario negatif atau nonideal	Berhasil

Berdasarkan Tabel 3, seluruh 11 skenario fungsi utama dapat dijalankan sesuai dengan rancangan aplikasi. Pengguna dapat memilih citra melalui galeri atau kamera, menjalankan inferensi menggunakan model TensorFlow Lite, melihat label prediksi dan confidence score, serta menyimpan hasil pemeriksaan ke dalam riwayat lokal. Fungsi navigasi, penyajian informasi, penampilan riwayat, dan penghapusan data juga berjalan sesuai dengan keluaran yang diharapkan.

Sebanyak 10 skenario negatif atau kondisi nonideal juga dapat ditangani tanpa menyebabkan aplikasi berhenti secara tidak terduga. Pada kondisi tanpa citra, berkas yang tidak dapat diproses, penolakan izin, atau model yang belum siap, aplikasi memberikan respons yang sesuai dan tidak melanjutkan proses inferensi secara tidak valid. Ketika riwayat belum memiliki data, aplikasi menampilkan kondisi riwayat kosong, sedangkan proses penghapusan seluruh riwayat dapat dijalankan tanpa mengganggu fungsi aplikasi lainnya.

Keberhasilan aplikasi dalam menerima citra buram, citra beresolusi rendah, atau citra berukuran besar hanya menunjukkan bahwa proses input dan inferensi dapat dijalankan secara fungsional. Hasil tersebut tidak menunjukkan bahwa ketepatan klasifikasi pada citra nonideal sama dengan citra normal. Pengaruh perubahan kualitas, resolusi, kompresi, pencahayaan, kontras, dan rotasi terhadap kinerja model dianalisis secara terpisah melalui pengujian robustness.

3.5 Hasil Evaluasi Kinerja Model

Evaluasi kinerja model dilakukan menggunakan 500 citra pada data test yang tidak digunakan dalam proses pelatihan, pemilihan model, maupun penentuan threshold. Data uji terdiri atas 250 citra otentik dan 250 citra AI-generate d. Pengujian dilakukan pada citra asli tanpa transformasi tambahan sebagai kondisi normal atau baseline untuk perbandingan dengan hasil pengujian robustness. Seluruh prediksi menggunakan threshold tetap sebesar 0,680 yang sebelumnya ditentukan menggunakan data validation. Hasil evaluasi model pada kondisi normal disajikan pada Tabel 4.

Tabel 4. Hasil Evaluasi MobileNetV2 pada Kondisi Normal

No.	Metrik	Nilai
1	Accuracy	85,40%
2	Precision weighted	85,53%
3	Recall weighted	85,40%
4	F1-score weighted	85,39%

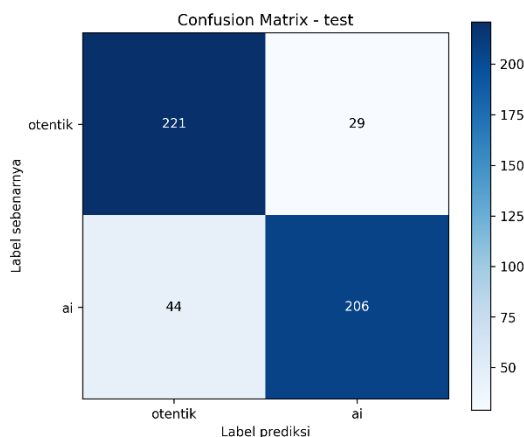
Berdasarkan Tabel 4, MobileNetV2 memperoleh accuracy sebesar 85,40%. Nilai tersebut menunjukkan bahwa 427 dari 500 citra uji berhasil diklasifikasikan sesuai dengan kelas aktual, sedangkan 73 citra lainnya mengalami kesalahan klasifikasi. Dengan demikian, tingkat kesalahan model pada kondisi normal adalah 14,60%.

Nilai precision weighted sebesar 85,53% menunjukkan tingkat ketepatan prediksi model setelah mempertimbangkan jumlah citra pada masing-masing kelas. Sementara itu, recall weighted sebesar 85,40% menunjukkan kemampuan model dalam mengenali sampel berdasarkan kelas aktualnya. Nilai F1-score weighted sebesar 85,39% memperlihatkan keseimbangan antara precision dan recall pada keseluruhan data uji.

Kedekatan nilai accuracy, precision, recall, dan F1-score menunjukkan bahwa performa model pada kondisi normal tidak hanya bergantung pada satu metrik. Namun, metrik agregat tersebut belum memperlihatkan jenis kesalahan yang terjadi pada masing-masing kelas. Oleh karena itu, distribusi prediksi benar dan salah dianalisis lebih lanjut menggunakan confusion matrix, False Positive, dan False Negative pada Subbab 3.6.

3.6 Confusion Matrix dan Analisis False Positive serta False Negative

Pengujian Analisis Confusion Matrix dilakukan untuk mengetahui distribusi prediksi benar dan salah pada setiap kelas secara lebih rinci. Dalam penelitian ini, citra AI-generate d ditetapkan sebagai kelas positif, sedangkan citra otentik ditetapkan sebagai kelas negatif. Pengujian menggunakan 500 citra pada data test yang terdiri atas 250 citra otentik dan 250 citra AI-generate d. Seluruh prediksi menggunakan threshold tetap sebesar 0,680. Hasil Confusion Matrix pada kondisi normal disajikan pada Gambar 4.



Gambar 4. Confusion Matrix Hasil Pengujian MobileNetV2

Berdasarkan Gambar 4, model berhasil mengklasifikasikan 221 citra otentik sebagai citra otentik atau true negative dan 206 citra AI-generate d sebagai citra AI-generate d atau true positive. Dengan demikian, terdapat 427 prediksi benar dari keseluruhan 500 citra uji. Nilai accuracy dihitung menggunakan persamaan berikut:

$$\text{Accuracy} = \frac{TN+TP}{TN+FP+FN+TP} \times 100\%$$

$$\text{Accuracy} = \frac{221+206}{500} \times 100\% = 85,40\%$$

Kesalahan klasifikasi terdiri atas 29 False Positive, yaitu citra otentik yang salah diprediksi sebagai citra AI-generate d, dan 44 False Negative, yaitu citra AI-generate d yang salah diprediksi sebagai citra otentik. Dari 250 citra otentik, sebanyak 29 citra atau 11,60% mengalami kesalahan klasifikasi. Sementara itu, sebanyak 44 dari 250 citra AI-generate d atau 17,60% tidak berhasil dikenali sebagai citra sintesis.

Model menghasilkan 235 prediksi kelas AI-generate d, yang berasal dari 206 true positive dan 29 False Positive. Model juga menghasilkan 265 prediksi kelas otentik, yang terdiri atas 221 true negative dan 44 False Negative. Jumlah False Negative yang lebih tinggi daripada False Positive menunjukkan bahwa model lebih sering gagal mengenali citra AI-generate d dan memprediksinya sebagai citra otentik. Hal tersebut juga terlihat dari tingkat pengenalan kelas AI-generate d sebesar 82,40%, yang lebih rendah dibandingkan tingkat pengenalan kelas otentik sebesar 88,40%.

Kesalahan False Negative perlu diperhatikan karena dapat menyebabkan citra AI-generate d dianggap sebagai citra otentik. Sementara itu, False Positive dapat menyebabkan pengguna meragukan citra yang sebenarnya berasal dari proses perekaman nyata. Oleh karena itu, hasil prediksi dalam aplikasi disertai dengan confidence score dan informasi mengenai keterbatasan model. Keluaran aplikasi digunakan sebagai indikasi awal untuk membantu pemeriksaan citra dan tidak diposisikan sebagai keputusan verifikasi forensik final.

3.7 Hasil Pengujian Robustness

Pengujian robustness dilakukan untuk mengetahui perubahan kinerja MobileNetV2 ketika citra mengalami transformasi yang dapat terjadi selama penyimpanan, penyuntingan, dan distribusi digital. Pengujian menggunakan salinan 500 citra pada data test yang sama dengan evaluasi kondisi normal, terdiri atas 250 citra otentik dan 250 citra AI-generate d. Seluruh skenario menggunakan model TensorFlow Lite dan threshold tetap sebesar 0,680 tanpa pelatihan ulang maupun perubahan label. Hasil pengujian disajikan pada Tabel 5.

Tabel 5. Hasil Pengujian Robustness MobileNetV2

Skenario	Accuracy	Precision	Recall	F1-score	Δ Accuracy
Normal	85,40%	85,53%	85,40%	85,39%	0,00
JPEG q=65	86,00%	86,15%	86,00%	85,99%	+0,60
JPEG q=30	86,40%	86,93%	86,40%	86,35%	+1,00
Resize 112 × 112	73,20%	79,57%	73,20%	71,68%	-12,20
Resize 64 × 64	63,00%	74,95%	63,00%	57,97%	-22,40

Skenario	Accuracy	Precision	Recall	F1-score	Δ Accuracy
Resize 112 × 112 + JPEG q=65	62,40%	76,38%	62,40%	56,66%	−23,00
Brightness 0,7	81,60%	83,40%	81,60%	81,35%	−3,80
Brightness 1,3	86,20%	86,25%	86,20%	86,20%	+0,80
Contrast 0,7	81,00%	83,25%	81,00%	80,67%	−4,40
Contrast 1,3	87,00%	87,13%	87,00%	86,99%	+1,60
Rotation −10°	84,40%	84,97%	84,40%	84,34%	−1,00
Rotation +10°	86,60%	87,33%	86,60%	86,53%	+1,20

Berdasarkan Tabel 5, model relatif mampu mempertahankan kinerja terhadap kompresi JPEG dan rotasi ringan. JPEG q=65 dan q=30 masing-masing menghasilkan accuracy 86,00% dan 86,40%, sedangkan rotation −10° dan +10° menghasilkan 84,40% dan 86,60%. Perubahan brightness dan contrast memberikan hasil yang beragam. Penurunan brightness dan contrast menjadi 0,7 menurunkan accuracy menjadi 81,60% dan 81,00%, sedangkan peningkatan contrast menjadi 1,3 menghasilkan nilai tertinggi sebesar 87,00%. Peningkatan tersebut hanya menggambarkan perubahan prediksi pada data uji penelitian ini dan tidak menunjukkan bahwa transformasi selalu meningkatkan performa model.

Penurunan terbesar terjadi pada perubahan resolusi. Resize 112 × 112 menghasilkan accuracy 73,20%, sedangkan resize 64 × 64 menurunkannya menjadi 63,00%. Kondisi terendah diperoleh pada kombinasi resize 112 × 112 dan JPEG q=65 dengan accuracy 62,40% dan F1-score 56,66%. Pada skenario tersebut, model hanya berhasil mengenali 65 dari 250 citra AI-generate d, sementara 185 citra lainnya salah diprediksi sebagai citra otentik.

Secara keseluruhan, MobileNetV2 lebih tahan terhadap kompresi JPEG, rotasi ringan, dan perubahan intensitas tertentu dibandingkan terhadap penurunan resolusi. Hasil ini menunjukkan bahwa pengurangan ukuran citra, terutama ketika disertai kompresi, menghilangkan informasi visual yang dibutuhkan model untuk mengenali citra AI-generate d. Oleh karena itu, hasil deteksi pada citra beresolusi rendah perlu ditafsirkan secara hati-hati.

3.8 Hasil Pengujian Performa On-device

Pengujian Pengujian performa on-device dilakukan untuk mengetahui kemampuan aplikasi menjalankan model TensorFlow Lite secara langsung pada perangkat Android. Pengujian menggunakan dua perangkat dengan spesifikasi berbeda. Model, citra masukan, konfigurasi threshold sebesar 0,680, dan tahapan pra-pemrosesan dibuat sama agar hasil kedua perangkat dapat dibandingkan. Parameter yang diamati meliputi rata-rata waktu inferensi, keberhasilan menampilkan hasil prediksi, dan kestabilan aplikasi. Ringkasan hasil pengujian disajikan pada Tabel 6.

Tabel 6. Hasil Pengujian Performa On-device

No.	Perangkat	Spesifikasi Utama	Sistem Operasi	Rata-rata Inferensi	Status
1	Realme 7 Pro	RAM 8 GB, Snapdragon 720G	Android 12	181,97 ms	Stabil
2	Xiaomi Redmi Note 9	RAM 6 GB, Helio G85	Android 10	263,62 ms	Stabil

Berdasarkan Tabel 6, aplikasi berhasil menjalankan inferensi secara lokal pada kedua perangkat. Realme 7 Pro memperoleh rata-rata waktu inferensi sebesar 181,97 ms, sedangkan Xiaomi Redmi Note 9 memperoleh 263,62 ms. Selisih waktu inferensi kedua perangkat adalah 81,65 ms, dengan Realme 7 Pro memberikan waktu eksekusi yang lebih cepat. Perbedaan tersebut menunjukkan bahwa spesifikasi perangkat dapat memengaruhi kecepatan pelaksanaan model TensorFlow Lite.

Aplikasi tetap dapat memuat model, menjalankan inferensi, serta menampilkan label prediksi dan confidence score pada kedua perangkat tanpa berhenti secara tidak terduga selama pengujian. Waktu inferensi pada kedua perangkat berada di bawah 300 ms untuk pemrosesan satu citra. Namun, nilai tersebut hanya merepresentasikan waktu eksekusi model dan tidak mencakup keseluruhan waktu pemilihan citra, pra-pemrosesan, pembaruan antarmuka, dan penyimpanan riwayat.

Pengujian masih terbatas pada dua perangkat Android sehingga hasilnya belum dapat digeneralisasikan untuk seluruh perangkat mobile. Perangkat dengan prosesor, kapasitas memori, dan versi sistem operasi yang berbeda dapat menghasilkan waktu inferensi yang berbeda. Pengujian selanjutnya dapat melibatkan lebih banyak perangkat serta menambahkan pengukuran penggunaan memori, konsumsi daya, dan suhu perangkat.

3.9 Pembahasan

Hasil Hasil penelitian menunjukkan bahwa MobileNetV2 dapat diterapkan pada aplikasi Android melalui TensorFlow Lite untuk melakukan deteksi awal citra otentik dan citra AI-generate d. Aplikasi mampu menerima citra dari galeri atau kamera, melakukan pra-pemrosesan dan inferensi secara on-device, menampilkan label serta confidence score, dan menyimpan riwayat secara lokal. Pengujian fungsional menunjukkan bahwa 11 fungsi utama

berjalan sesuai rancangan, sedangkan 10 skenario negatif atau nonideal dapat ditangani tanpa menyebabkan aplikasi berhenti secara tidak terduga.

Pada kondisi normal, model memperoleh accuracy 85,40%, precision weighted 85,53%, recall weighted 85,40%, dan F1-score weighted 85,39%. Sebanyak 427 dari 500 citra uji berhasil diklasifikasikan dengan benar, sedangkan 73 citra mengalami kesalahan klasifikasi. Hasil tersebut menunjukkan bahwa model mampu mengenali sebagian besar data uji, tetapi tingkat kesalahan 14,60% masih perlu diperhatikan. Tantangan generalisasi terhadap sumber citra dan generator yang beragam juga masih ditemukan pada penelitian deteksi citra sintetis sebelumnya [6], [9].

Hasil Confusion Matrix terdiri atas 221 true negative, 206 true positive, 29 False Positive, dan 44 False Negative. Model mampu mengenali 88,40% citra otentik dan 82,40% citra AI-generate d. Jumlah False Negative yang lebih tinggi menunjukkan bahwa model lebih sering memprediksi citra AI-generate d sebagai citra otentik. Oleh karena itu, hasil aplikasi perlu disertai confidence score dan informasi keterbatasan agar tidak digunakan sebagai keputusan forensik final.

Pengujian robustness menunjukkan bahwa model relatif mampu mempertahankan performa terhadap kompresi JPEG dan rotasi ringan. JPEG q=65 dan q=30 menghasilkan accuracy sebesar 86,00% dan 86,40%, sedangkan rotation -10° dan $+10^\circ$ menghasilkan 84,40% dan 86,60%. Perubahan brightness dan contrast memberikan hasil yang lebih beragam. Brightness 0,7 dan contrast 0,7 menurunkan accuracy menjadi 81,60% dan 81,00%, sedangkan brightness 1,3 dan contrast 1,3 menghasilkan 86,20% dan 87,00%. Peningkatan pada beberapa skenario hanya menggambarkan perubahan prediksi pada subset uji yang digunakan dan tidak menunjukkan bahwa transformasi selalu meningkatkan performa model. Pengaruh perubahan kualitas citra terhadap detektor juga dilaporkan pada penelitian sebelumnya [7], [15].

Penurunan terbesar terjadi pada transformasi resize. Ukuran 112×112 piksel menghasilkan accuracy 73,20%, sedangkan ukuran 64×64 piksel menghasilkan 63,00%. Kondisi terendah diperoleh pada kombinasi resize 112×112 dan JPEG q=65 dengan accuracy 62,40% dan F1-score 56,66%. Pada skenario tersebut, model hanya mengenali 65 dari 250 citra AI-generate d, sementara 185 citra lainnya salah diprediksi sebagai citra otentik. Hasil tersebut menunjukkan bahwa pengurangan resolusi dapat menghilangkan detail dan tekstur yang diperlukan model dalam membedakan kedua kelas. Temuan ini sejalan dengan penelitian yang menunjukkan bahwa perubahan resolusi dan kondisi distribusi dunia nyata dapat menurunkan keandalan detektor [16], [17].

Dari sisi performa on-device, aplikasi memperoleh rata-rata waktu inferensi 181,97 ms pada Realme 7 Pro dan 263,62 ms pada Xiaomi Redmi Note 9. Perbedaan tersebut menunjukkan bahwa spesifikasi perangkat memengaruhi kecepatan eksekusi model. Meskipun demikian, aplikasi tetap dapat menjalankan inferensi dan menampilkan hasil secara stabil pada kedua perangkat. Hasil ini mendukung penggunaan MobileNetV2 sebagai arsitektur yang efisien serta TensorFlow Lite sebagai format implementasi pada perangkat mobile [22], [23].

Secara keseluruhan, penelitian ini berhasil mengintegrasikan model deteksi citra AI-generate d ke dalam aplikasi Android dan mengevaluasi ketahanannya terhadap beberapa transformasi citra. Model relatif stabil terhadap JPEG, rotasi ringan, dan beberapa perubahan intensitas, tetapi sensitif terhadap penurunan resolusi. Keterbatasan penelitian meliputi jumlah dataset sebanyak 2.000 citra, komposisi generator yang belum sepenuhnya merata, penggunaan satu arsitektur model, dan pengujian pada dua perangkat Android. Penelitian selanjutnya dapat memperluas sumber data, menambahkan generator baru, meningkatkan ketahanan terhadap citra beresolusi rendah, serta menguji model pada lebih banyak perangkat.

4. KESIMPULAN

Penelitian ini berhasil mengimplementasikan MobileNetV2 dalam aplikasi Android untuk melakukan deteksi awal citra otentik dan citra AI-generate d secara on-device menggunakan TensorFlow Lite. Aplikasi mampu menerima citra melalui galeri atau kamera, melakukan pra-pemrosesan dan inferensi lokal, menampilkan label prediksi serta confidence score, dan menyimpan riwayat deteksi secara lokal. Pengujian fungsional menunjukkan bahwa 11 fungsi utama berjalan sesuai rancangan, sedangkan 10 skenario negatif atau nonideal dapat ditangani tanpa menyebabkan aplikasi berhenti secara tidak terduga. Pada pengujian terhadap 500 citra kondisi normal dengan threshold 0,680, model memperoleh accuracy 85,40%, precision weighted 85,53%, recall weighted 85,40%, dan F1-score weighted 85,39%. Hasil Confusion Matrix terdiri atas 221 true negative, 206 true positive, 29 False Positive, dan 44 False Negative, yang menunjukkan bahwa model lebih sering gagal mengenali citra AI-generate d sebagai citra sintetis dibandingkan salah menandai citra otentik. Pengujian robustness menunjukkan bahwa model relatif mampu mempertahankan performa terhadap kompresi JPEG, rotasi ringan, dan beberapa perubahan intensitas citra. Nilai tertinggi pada skenario transformasi diperoleh pada contrast 1,3 dengan accuracy 87,00%, sedangkan kondisi terendah terjadi pada kombinasi resize 112×112 piksel dan JPEG q=65 dengan accuracy 62,40% serta F1-score 56,66%. Hasil tersebut menunjukkan bahwa penurunan resolusi memberikan dampak terbesar terhadap kemampuan model dalam mengenali citra AI-generate d. Pengujian on-device menghasilkan rata-rata waktu inferensi 181,97 ms pada Realme 7 Pro dan 263,62 ms pada Xiaomi Redmi Note 9, dengan aplikasi berjalan stabil pada kedua perangkat. Dengan demikian, aplikasi layak digunakan sebagai alat bantu deteksi awal yang praktis, tetapi belum dapat dijadikan alat verifikasi forensik final. Penelitian selanjutnya perlu memperluas



sumber dataset dan generator, meningkatkan ketahanan terhadap citra beresolusi rendah, membandingkan arsitektur ringan lainnya, serta melibatkan lebih banyak perangkat Android.

REFERENCES

- [1] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, “High-Resolution Image Synthesis with Latent Diffusion Models,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 10684–10695. doi: 10.1109/CVPR52688.2022.01042.
- [2] M. Zhu et al., “GenImage: A Million-Scale Benchmark for Detecting AI-generated Image,” *Adv. Neural Inf. Process. Syst.*, vol. 36, no. NeurIPS, 2023.
- [3] S. J. Nightingale and H. Farid, “AI-Synthesized Faces Are Indistinguishable from Real Faces and More Trustworthy,” *Proc. Natl. Acad. Sci.*, vol. 119, no. 8, p. e2120481119, 2022, doi: 10.1073/pnas.2120481119.
- [4] J. J. Bird and A. Lotfi, “CIFAKE: Image Classification and Explainable Identification of AI-generated Synthetic Images,” *IEEE Access*, vol. 12, pp. 26896–26909, 2024, doi: 10.1109/ACCESS.2024.3356122.
- [5] S. S. Baraheem and T. V. Nguyen, “AI vs. AI: Can AI Detect AI-generated Images?,” *J. Imaging*, vol. 9, no. 10, p. 199, 2023, doi: 10.3390/jimaging9100199.
- [6] U. Ojha, Y. Li, and Y. J. Lee, “Towards Universal Fake Image Detectors that Generalize Across Generative Models,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2023-June, pp. 24480–24489, 2023, doi: 10.1109/CVPR52729.2023.02345.
- [7] R. Corvi, D. Cozzolino, G. Zingarini, G. Poggi, K. Nagano, and L. Verdoliva, “On the Detection of Synthetic Images Generated by Diffusion Models,” 2023, IEEE. doi: 10.1109/ICASSP49357.2023.10095167.
- [8] Z. Wang et al., “DIRE for Diffusion-Generated Image Detection,” pp. 22445–22455, 2023, [Online]. Available: https://openaccess.thecvf.com/content/ICCV2023/html/Wang_DIRE_for_Diffusion-Generated_Image_Detection_ICCV_2023_paper.html
- [9] D. Cozzolino, G. Poggi, R. Corvi, M. Nießner, and L. Verdoliva, “Raising the Bar of AI-generated Image Detection with CLIP,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2024, pp. 4356–4366. doi: 10.1109/CVPRW63382.2024.00439.
- [10] C. Tan, Y. Zhao, S. Wei, G. Gu, P. Liu, and Y. Wei, “Rethinking the Up-sampling Operations in CNN-Based Generative Network for Generalizable Deepfake Detection,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024, pp. 28130–28139. doi: 10.1109/CVPR52733.2024.02657.
- [11] Y. Li et al., “MaskSim: Detection of Synthetic Images by Masked Spectrum Similarity Analysis,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2024, pp. 3855–3865. doi: 10.1109/CVPRW63382.2024.00390.
- [12] J. Ricker, D. Lukovnikov, and A. Fischer, “AEROBLADE: Training-Free Detection of Latent Diffusion Images Using Autoencoder Reconstruction Error,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024, pp. 9130–9140. doi: 10.1109/CVPR52733.2024.00872.
- [13] H. Zhang, Q. He, X. Bi, W. Li, B. Liu, and B. Xiao, “Towards Universal AI-generated Image Detection by Variational Information Bottleneck Network,” pp. 23828–23837, 2025, doi: 10.1109/cvpr52734.2025.02219.
- [14] F. Guillaro et al., “A Bias-Free Training Paradigm for More General AI-generated Image Detection Real,” pp. 18685–18694.
- [15] Q. Bammey, “Synthbuster: Towards Detection of Diffusion Model Generated Images,” *IEEE Open J. Signal Process.*, vol. 5, pp. 1–9, 2024, doi: 10.1109/OJSP.2023.3337714.
- [16] C. Li et al., “RRDataset: Bridging the Gap Between Ideal and Real-world Evaluation: Benchmarking AI-generated Image Detection in Challenging Scenarios,” pp. 20379–20389, 2025, [Online]. Available: <http://arxiv.org/abs/2509.09172>
- [17] D. Karageorgiou, S. Papadopoulos, I. Kompatsiaris, and E. Gavves, “Any-Resolution AI-generated Image Detection by Spectral Learning,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025, pp. 18706–18717. doi: 10.1109/CVPR52734.2025.01743.
- [18] J. Park and A. Owens, “Community Forensics: Using Thousands of Generators to Train Fake Image Detectors,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025, pp. 8245–8257. doi: 10.1109/CVPR52734.2025.00772.
- [19] Z. Jia et al., “Secret Lies in Color: Enhancing AI-generated Images Detection with Color Distribution Analysis,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025, pp. 13445–13454. doi: 10.1109/CVPR52734.2025.01255.
- [20] B. Chu, X. Xu, X. Wang, Y. Zhang, W. You, and L. Zhou, “FIRE: Robust Detection of Diffusion-Generated Images via Frequency-Guided Reconstruction Error,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025, pp. 12830–12839. doi: 10.1109/CVPR52734.2025.01197.
- [21] H. Yu and B. Xu, “Multi-Modal Texture Fusion Network for Detecting AI-generated Images,” *Front. Artif. Intell.*, vol. 8, p. 1663292, 2025, doi: 10.3389/frai.2025.1663292.
- [22] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen, “MobileNetV2: Inverted Residuals and Linear Bottlenecks,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 4510–4520, 2018, doi: 10.1109/CVPR.2018.00474.
- [23] R. David et al., “TensorFlow Lite Micro: Embedded Machine Learning for TinyML Systems,” in *Proceedings of Machine Learning and Systems*, 2021, pp. 800–811. doi: 10.48550/arXiv.2010.08678.