

Sistem Deteksi Citra Wajah Manusia Hasil Generatif AI Berbasis Mobile Menggunakan Model Transfer Learning MobileNetV2

Miftakh Damar Faizin¹, Asmunin²

Program Studi Manajemen Informatika, Universitas Negeri Surabaya
Jl. Ketintang, Ketintang, Kec. Gayungan, Kota Surabaya, Jawa Timur 60231

¹miftakhdamar.22013@mhs.unesa.ac.id

²asmunin@unesa.ac.id

Abstrak - Kemajuan AI generatif memicu risiko penyalahgunaan identitas digital melalui pembuatan gambar wajah sintetis. Untuk memitigasi risiko tersebut, penelitian ini mengembangkan sistem deteksi gambar wajah hasil AI berbasis aplikasi mobile menggunakan arsitektur MobileNetV2. Arsitektur training model menggunakan Adam optimizer dengan learning rate 5e-6. Hasil penelitian menunjukkan bahwa model MobileNetV2 dengan Adam optimizer mencapai akurasi sebesar 99% pada data validasi dan data testing. Model tersebut kemudian dikonversi ke format TensorFlow Lite dan diintegrasikan ke dalam aplikasi mobile lintas platform berbasis Flutter dengan penyimpanan lokal SQLite. Pengujian inferensi secara langsung pada perangkat mobile dengan data real world berhasil mengklasifikasikan 35 dari 39 gambar uji dari berbagai sumber dan kondisi secara tepat, dengan 4 kesalahan dan 3 diantaranya pada gambar dengan kondisi wajah berjarak jauh. Sistem ini terbukti secara fungsional efektif dan praktis untuk mendeteksi manipulasi wajah secara lokal

Kata kunci: MobileNetV2, Generatif AI, Deteksi Wajah, Aplikasi Mobile, Adam Optimizer

Abstract - The advancement of generative AI has escalated the risk of digital identity misuse through the creation of synthetic facial images. To mitigate this risk, this study develops a mobile application-based AI-generated face detection system utilizing the MobileNetV2 architecture. The model training process employs the Adam optimizer with a learning rate of 5e-6. The research results indicate that the MobileNetV2 model with the Adam optimizer achieves an accuracy of 99% on both validation and testing datasets. Subsequently, the model was converted into the TensorFlow Lite format and integrated into a cross-platform mobile application built with Flutter, utilizing SQLite for local storage. Direct inference testing on mobile devices using real-world data successfully and accurately classified 35 out of 39 test images from various sources and conditions, resulting in 4 misclassifications, 3 of which occurred on images featuring subjects at a far distance. This system is proven to be functionally effective and practical for local facial manipulation.

Keywords: MobileNetV2, Generative AI, Face Detection, Mobile Application, Adam Optimizer

I. PENDAHULUAN

Perkembangan pesat model generatif seperti GAN (*Generative Adversarial Networks*) dan Diffusion Models menyebabkan munculnya citra wajah sintetis yang kualitasnya begitu realistis hingga tidak dapat dibedakan oleh mata manusia. Studi menunjukkan bahwa lebih dari 50% responden gagal membedakan wajah generatif AI (*Artificial Intelligence*) dari wajah asli, dan ironisnya, wajah hasil generatif AI dinilai lebih meyakinkan dibanding wajah asli [1]. Model seperti StyleGAN2 dan StyleGAN3 terbukti mampu menghasilkan citra wajah sintetis beresolusi tinggi yang secara statistik menyerupai distribusi wajah nyata, sehingga memunculkan risiko serius seperti penciptaan identitas palsu, akun bot yang tampak manusia, serta penyalahgunaan dalam verifikasi identitas digital [2][3][4][5][6]. Penelitian lain mengungkapkan bahwa wajah generatif AI juga berhasil digunakan untuk mendaftarkan akun palsu pada sistem autentikasi otomatis karena pola teksturnya semakin menyerupai wajah manusia [7]. Oleh karena itu, dibutuhkan sistem pendeteksi wajah sintetis yang tidak hanya akurat, tetapi juga mampu memberikan penjelasan yang jelas terhadap hasil deteksi dan juga dapat diakses dengan mudah oleh masyarakat luas melalui perangkat pribadi mereka [8][9][9].

Tantangan utama dalam mengintegrasikan sistem deteksi citra wajah hasil generatif AI ke dalam perangkat mobile terletak pada keterbatasan sumber daya komputasi, seperti kecepatan prosesor dan kapasitas RAM (*Random Access Memory*). Pada perangkat mobile, beban komputasi yang berlebihan dapat menyebabkan latensi yang tinggi, penggunaan memori yang berlebihan, peningkatan suhu, dan konsumsi daya yang meningkat. Untuk mengatasi kendala tersebut, penerapan arsitektur MobileNetV2 menjadi solusi yang sangat potensial. MobileNetV2 dirancang secara khusus sebagai model yang ringan [10]. Arsitektur ini efisien, memungkinkan proses deteksi dijalankan langsung di perangkat tanpa perlu memanfaatkan server eksternal, sekaligus menjaga konsumsi daya tetap rendah. Dengan demikian,

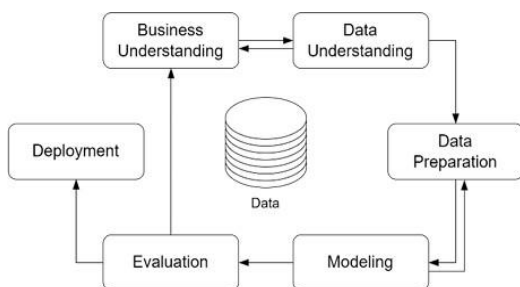
MobileNetV2 menjadi pilihan ideal untuk diintegrasikan pada aplikasi mobile.

Penelitian terbaru memperkuat bukti efektivitas model *transfer learning* MobileNetV2 dalam mendeteksi citra wajah hasil generatif AI. Berdasarkan penelitian ditemukan bahwa penggunaan MobileNetV2 sebagai model dasar mampu mencapai akurasi hingga 95% [11]. Dengan demikian, MobileNetV2 memberikan keseimbangan paling ideal antara akurasi, kecepatan, dan kebutuhan komputasi, sehingga sangat sesuai untuk diterapkan pada aplikasi *mobile* yang mengharuskan proses deteksi dilakukan secara lokal, cepat, dan tanpa ketergantungan pada server *eksternal*.

Berdasarkan landasan yang telah dijelaskan, penelitian ini berfokus pada pengembangan sistem deteksi wajah hasil generatif AI menggunakan MobileNetV2 yang diimplementasikan dalam bentuk aplikasi *mobile* agar dapat diakses secara luas oleh pengguna. Sistem yang dirancang diharapkan tidak hanya memiliki tingkat akurasi tinggi, tetapi juga mampu menyesuaikan diri terhadap variasi citra yang sering dijumpai di dunia nyata. Selain itu, hasil deteksi akan disajikan secara visual untuk membantu pengguna memahami proses dan keputusan sistem secara lebih intuitif. Dengan demikian, penelitian ini diharapkan dapat memberikan kontribusi nyata dalam mengurangi penyebaran konten visual palsu serta menjaga integritas informasi digital di era perkembangan generatif AI.

II. METODE PENELITIAN

CRISP-DM (*Cross Industry Standard Process for Data Mining*) digunakan sebagai metode penelitian ini karena menggunakan alur kerja yang terstruktur dan cocok untuk penelitian ini.



Gambar. 1 Diagram Alur *CRISP-DM* [12]

Pada Gambar. 1 terdapat tahapan metode *CRISP-DM* yang terdiri 6 tahapan yaitu *Business Understanding*, *Data Understanding*, *Data Preparation*, *Modeling*, *Evaluation*, dan *Deployment*.

A. Business Understanding

Meningkatnya penyebaran citra wajah sintesis hasil *Generative AI* menimbulkan risiko terhadap penipuan identitas digital maupun berbagai bentuk kejahatan di ruang siber lainnya [9]. Oleh karena itu, penelitian ini bertujuan mengembangkan sistem yang mampu mengklasifikasikan apakah suatu wajah merupakan wajah

asli atau hasil generatif AI serta dapat diakses dengan mudah melalui aplikasi *mobile*. Pengembangan aplikasi *mobile* dipilih agar proses deteksi dapat dilakukan secara praktis kapan saja dan di mana saja hanya dengan menggunakan perangkat *smartphone*.

B. Data Understanding

Dataset diambil dari platform Kaggle dengan judul *dataset* “RealVsFake_162k_by_Wish” yang diunggah oleh “Wish096” pada tahun 2025.

TABEL 1
DISTRIBUSI *DATASET*

Segmentasi Data	Jumlah Gambar	Distribusi Jenis Gambar
<i>Real</i>	81.000	a. FFHQ: 40.500 b. CelebA: 40.500
<i>Fake</i>	80.972	a. AiGenImage: 1.001 b. StyleGAN: 70.000 c. Stable Diffusion: 9.971

Pada Tabel 1 terdapat rincian distribusi dataset yakni Jumlah data gambar *real* sebanyak 81.000 dan jumlah data gambar *Fake* sebanyak 80.972. Pada setiap segmen, data dibagi menjadi 70% untuk *training*, 15% untuk *validation* dan 15% untuk *testing*.

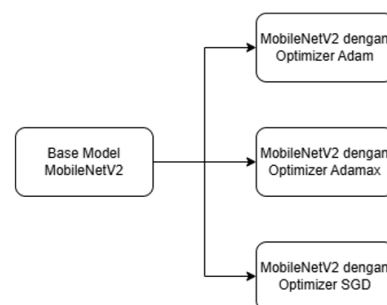
C. Data Preparation



Gambar. 2 Alur *Preprocessing* Data

Tahap *data preparation* dilakukan untuk meningkatkan kualitas data sebelum proses pelatihan model. Pada Gambar. 2 terdapat alur *preprocessing* data yakni setiap citra terlebih dahulu diproses menggunakan *BlazeFace* untuk mendeteksi area wajah, kemudian dilakukan *cropping* sehingga hanya bagian wajah yang digunakan sebagai masukan model. Selanjutnya dilakukan *resize* sesuai ukuran input MobileNetV2 yaitu 224x224 piksel serta augmentasi data berupa rotasi, *zoom*, dan *horizontal flip* untuk meningkatkan variasi data pelatihan dan mengurangi risiko *overfitting*.

D. Modelling

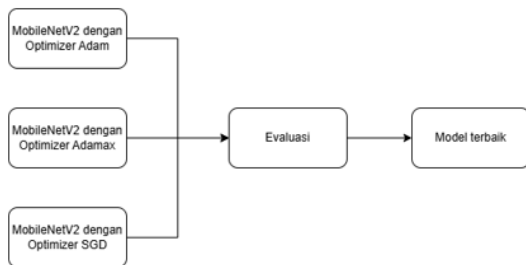


Gambar. 3 Alur Pemodelan

Model klasifikasi dibangun menggunakan pendekatan *transfer learning* dengan MobileNetV2 sebagai *base model*. Pada Gambar. 3 terdapat alur pemodelan yakni Selama proses pelatihan dan pengujian model, dilakukan analisis komparatif terhadap tiga jenis *optimizer* yang berbeda, yaitu Adam, Adamax, dan SGD. Untuk menjamin keadilan dalam pengujian, ketiga *optimizer* tersebut dievaluasi menggunakan konfigurasi parameter pelatihan yang sepenuhnya sama, termasuk pada penentuan nilai *learning rate*, jumlah *epoch*, dan ukuran *batch*.

E. Evaluation

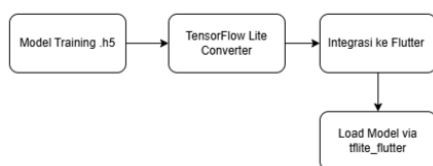
Evaluasi model dilakukan menggunakan data uji dengan membentuk *confusion matrix* yang terdiri atas *True Positive (TP)*, *True Negative (TN)*, *False Positive (FP)*, dan *False Negative (FN)*. Berdasarkan *confusion matrix* tersebut dihitung beberapa metrik evaluasi, yaitu *accuracy*, *precision*, *recall*, dan *F1-score*. *Accuracy* digunakan untuk mengukur proporsi keseluruhan prediksi yang benar terhadap seluruh data uji. *Precision* mengukur ketepatan model dalam mengidentifikasi citra wajah hasil AI generatif sehingga menunjukkan seberapa sedikit prediksi positif yang salah. *Recall* mengukur kemampuan model dalam menemukan seluruh citra wajah hasil AI generatif yang sebenarnya ada pada data uji. Sementara itu, *F1-score* merupakan rata-rata harmonis antara *precision* dan *recall* sehingga memberikan ukuran performa yang lebih seimbang, terutama ketika distribusi kelas tidak sepenuhnya seimbang.



Gambar. 4 Alur Perbandingan Model

Pada Gambar. 4 terdapat alur perbandingan model yakni Hasil evaluasi dari ketiga model dianalisis dan dibandingkan guna mengetahui pengaruh *optimizer* terhadap peningkatan performa model. Model yang menunjukkan performa terbaik berdasarkan nilai metrik evaluasi tersebut, khususnya dalam hal keseimbangan antara ketepatan dan kemampuan deteksi, selanjutnya dipilih untuk diintegrasikan ke dalam aplikasi *mobile* berbasis Flutter sebagai model inferensi utama.

F. Deployment



Gambar. 5 Alur Integrasi Model

Pada Gambar 5 ditampilkan alur integrasi model ke dalam aplikasi *mobile*. Setelah melalui tahapan evaluasi, model dengan performa terbaik dari hasil pelatihan kemudian dikonversi ke format *TensorFlow Lite (TFLite)* guna mereduksi ukuran file dan mengoptimalkan efisiensi komputasinya agar kompatibel dengan perangkat seluler. Model TFLite yang telah dioptimalkan tersebut kemudian diintegrasikan ke dalam aplikasi *mobile* yang dikembangkan menggunakan *framework* Flutter. Dengan mengadopsi sistem ini, seluruh proses inferensi atau eksekusi model dapat dijalankan sepenuhnya secara lokal di dalam perangkat, sehingga aplikasi mampu melakukan klasifikasi citra wajah secara responsif dan *real-time* tanpa bergantung pada koneksi internet maupun infrastruktur *server* eksternal. Pendekatan ini juga memberikan keuntungan berupa peningkatan privasi data pengguna.

III. HASIL DAN PEMBAHASAN

A. Hasil Persiapan Data

Seluruh gambar mentah pada *dataset* diproses menggunakan algoritma MediaPipe BlazeFace untuk mendeteksi koordinat wajah. Algoritma diatur menggunakan ambang batas (*threshold*) minimal 0.5 untuk tingkat keyakinan (*confidence*) deteksi. Area wajah yang berhasil dideteksi kemudian dipotong mengikuti *output* titik koordinat dari MediaPipe BlazeFace untuk meminimalkan gangguan dari latar belakang (*background noise*). Setelah dipotong, gambar diubah ukurannya (*resizing*) menjadi 224x224 piksel untuk memenuhi standar lapisan masukan (input layer) pada arsitektur MobileNetV2.



Gambar. 6 Dataset Sebelum *Preprocessing*



Gambar. 7 Dataset Sesudah *Preprocessing*

Pada Gambar. 6 ditampilkan contoh citra sebelum dilakukan tahap *preprocessing*. Berdasarkan gambar tersebut, dapat dilihat bahwa objek wajah belum menjadi fokus utama karena masih terdapat *noise* pada bagian *background*, sehingga informasi yang tidak relevan masih ikut terekam. Sementara itu, Gambar. 7 menampilkan hasil citra setelah melalui proses *preprocessing*, di mana area wajah telah berhasil dipusatkan sebagai objek utama dan sebagian besar *background* yang tidak diperlukan telah dihilangkan. Proses ini bertujuan untuk mengurangi gangguan pada citra serta membantu model dalam mengekstraksi fitur wajah secara lebih optimal selama proses pelatihan model.

TABEL 2
DISTRIBUSI DATASET SETELAH PREPROCESSING DATA

Kategori Data	Presentase	Jumlah Gambar
Data Training	70%	113.320 (56.638 <i>fake</i> dan 56.682 <i>real</i>)
Data Validation	15%	24.282 (12.136 <i>fake</i> dan 12.146 <i>real</i>)
Data Testing	15%	24.285 (12.138 <i>fake</i> dan 12.147 <i>real</i>)

Dataset gambar wajah yang telah melalui tahap *preprocessing* kemudian dibagi ke dalam tiga subset data: pelatihan (*training*), validasi (*validation*), dan pengujian (*testing*). Pada Tabel 2 terlihat pembagian dilakukan menggunakan *library* *splitfolders* dengan rasio 70:15:15 dan diatur menggunakan nilai *seed* 42 untuk menjaga konsistensi pengacakan parameter data.

TABEL 3
KONFIGURASI PARAMETER AUGMENTASI DATA TRAINING

Parameter	Nilai
Rescale	1./255
Rotation Range	90
Zoom Range	0.2
Brightness Range	[0.1, 1.1]
Horizontal Flip	True

Pada Tabel 3 terdapat parameter konfigurasi augmentasi data, proses ini dilakukan untuk meningkatkan keberagaman variasi gambar pada tahap pelatihan serta mencegah terjadinya *overfitting* pada kelompok data training. Proses ini memanfaatkan fungsi *ImageDataGenerator* yang secara otomatis memodifikasi gambar secara acak pada setiap iterasi pelatihan tanpa mengubah data aslinya di dalam penyimpanan. Parameter lengkap augmentasi terdapat pada Tabel 3, selain itu Proses augmentasi ini juga sekaligus menangani normalisasi nilai piksel (penskalaan 1./255) agar berada pada rentang 0 hingga 1.

B. Hasil Pemodelan

Tahap pemodelan bertujuan untuk melatih arsitektur MobileNetV2 dalam mengklasifikasikan gambar wajah asli dan gambar wajah buatan AI. Pada penelitian ini, dilakukan eksperimen menggunakan 3 skenario pemodelan berbasis arsitektur MobileNetV2 untuk membandingkan dan mencari performa yang paling optimal.

TABEL 4
HYPERPARAMETER PELATIHAN MODEL

Parameter	Nilai
Optimizer	Adam, Adamax, SGD
Learning Rate	5e-6
Batch Size	32
Epoch	200
Early Stopping	True, Patience 10 Epoch, Monitoring Validasi Loss
Save Best Only	True

Pada Tabel 4 dan terdapat Hyperparameter pelatihan model, proses pelatihan membagi dataset ke dalam kelompok-kelompok kecil dengan ukuran batch size sebesar 32 (nilai default) gambar yang diproses secara bersamaan dalam satu iterasi. Pelatihan awal model diatur sebanyak 200 epoch, kemudian diterapkan teknik *early stopping* dengan memantau nilai *validation loss* dan menetapkan nilai *patience*

sebesar 10 epoch. Dengan konfigurasi tersebut, apabila selama 10 epoch berturut-turut tidak terjadi penurunan *validation loss*, maka proses pelatihan akan dihentikan secara otomatis untuk mencegah terjadinya *overfitting* sekaligus menghemat waktu komputasi. Selain itu, digunakan pula parameter *save best only* sehingga hanya model dengan performa terbaik berdasarkan nilai *validation loss* yang akan disimpan. Penerapan kombinasi strategi tersebut bertujuan untuk memperoleh model yang memiliki kemampuan generalisasi yang baik terhadap data baru.

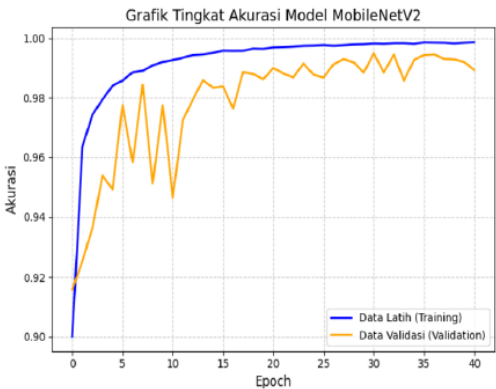
TABEL 5
EKOSISTEM PENGEMBANGAN MODEL

Parameter	Nilai
GPU	RTX 5060 Laptop 8 GB
CPU	Intel Core Ultra 7 255HX
Tools	Jupyter Notebook
Environment Manager	Anaconda

Pada Tabel 5 disajikan rincian ekosistem yang digunakan dalam proses pengembangan model. Informasi tersebut disertakan karena setiap ekosistem pengembangan yang digunakan, baik dari sisi perangkat keras maupun perangkat lunak, dapat memberikan pengaruh terhadap proses pelatihan model, sehingga berpotensi menghasilkan performa yang berbeda pada model yang dikembangkan

C. Hasil Evaluasi Model

Tahap evaluasi dilakukan untuk mengukur seberapa baik kinerja model dalam mengenali dan mengklasifikasikan gambar wajah asli maupun hasil AI. Evaluasi pada penelitian ini dibagi menjadi dua tahapan utama, yaitu evaluasi selama proses pelatihan dan evaluasi pengujian akhir menggunakan data testing.



Gambar. 8 Akurasi Training MobileNetV2 dengan Adam Optimizer

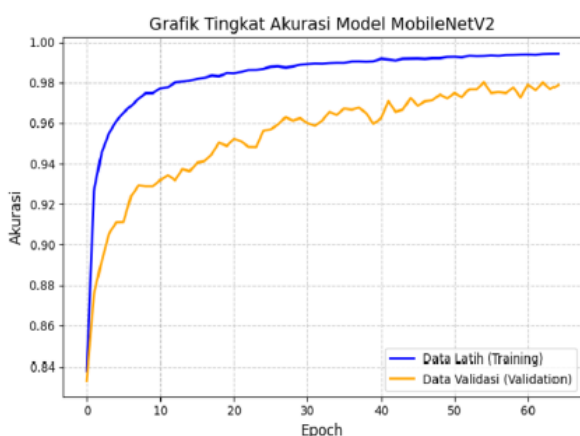
Hasil evaluasi proses pelatihan model yang dioptimasi dengan algoritma Adam dapat diamati pada Gambar. 8, di mana akurasi pelatihan berhasil menyentuh angka 99%. Penggunaan mekanisme *early stopping* menghentikan proses pembelajaran pada epoch ke-41, durasi komputasi keseluruhan untuk melatih model MobileNetV2 dengan adam optimizer ini membutuhkan waktu selama 6 jam 58 menit 49 detik.

Pada Gambar 9 ditampilkan hasil evaluasi kinerja model berbasis confusion matrix menggunakan data testing yang membuktikan bahwa implementasi optimizer Adam pada arsitektur MobileNetV2 mampu menghasilkan performa klasifikasi yang sangat baik. Model tersebut meraih tingkat keberhasilan yang konsisten di angka 99% untuk seluruh metrik pengujian, yakni akurasi, precision, recall, dan f1-score. Hasil ini menunjukkan bahwa model memiliki kemampuan yang sangat baik dalam membedakan citra fake dan real dengan tingkat kesalahan klasifikasi yang sangat rendah. Selain itu, konsistensi nilai pada seluruh metrik evaluasi mengindikasikan bahwa model tidak hanya memiliki akurasi yang tinggi, tetapi juga mampu mempertahankan keseimbangan antara kemampuan mendeteksi setiap kelas dan meminimalkan terjadinya false positive maupun false negative.

	precision	recall	f1-score	support
0	0.991	1.000	0.995	12137
1	1.000	0.991	0.995	12147
accuracy			0.995	24284
macro avg	0.995	0.995	0.995	24284
weighted avg	0.995	0.995	0.995	24284

Gambar. 9 Hasil Evaluasi Confusion Metrik pada Model dengan Adam Optimizer

Kinerja pelatihan model MobileNetV2 dengan optimizer Adamax dapat diamati pada grafik Gambar. 10, di mana tingkat akurasi berhasil mencapai angka 99%. Siklus pelatihan tersebut dihentikan oleh sistem early stopping pada epoch ke-65, dengan total waktu komputasi mencapai 10 jam 40 menit 9 detik. Durasi pemrosesan ini terbukti membutuhkan waktu yang lebih lama apabila dikomparasikan dengan kinerja arsitektur yang menggunakan Adam optimizer.



Gambar. 10 Akurasi Training MobileNetV2 dengan Adamax Optimizer

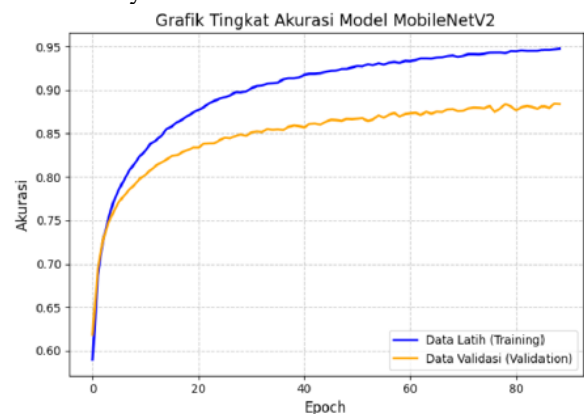
Pada Gambar 11 ditampilkan hasil pengujian performa klasifikasi yang dievaluasi melalui confusion matrix menggunakan data testing yang menunjukkan bahwa arsitektur MobileNetV2 dengan optimizer Adamax mampu mencapai kinerja yang sangat optimal. Model ini mencatatkan tingkat keberhasilan sebesar 98% secara merata pada indikator akurasi, precision, recall, maupun f1-score, sedikit lebih rendah jika dibandingkan dengan model yang menggunakan

optimizer Adam. Meskipun demikian, selisih performa yang hanya sebesar 1% menunjukkan bahwa Adamax tetap mampu menghasilkan model dengan kemampuan klasifikasi yang sangat baik dan konsisten. Hasil tersebut mengindikasikan bahwa optimizer Adamax masih layak digunakan sebagai alternatif dalam pengembangan sistem deteksi deepfake, terutama ketika mempertimbangkan karakteristik proses pelatihan dan kebutuhan implementasi model. Selain itu, tingginya nilai pada seluruh metrik evaluasi menunjukkan bahwa model mampu mempertahankan keseimbangan antara kemampuan mengenali citra pada masing-masing kelas dan meminimalkan kesalahan klasifikasi selama proses pengujian.

	precision	recall	f1-score	support
0	0.963	1.000	0.981	12137
1	1.000	0.961	0.980	12147
accuracy			0.981	24284
macro avg	0.981	0.981	0.981	24284
weighted avg	0.981	0.981	0.981	24284

Gambar. 11 Hasil Evaluasi Confusion Metrik pada Model dengan Adamax Optimizer

Kinerja pelatihan model yang menggunakan algoritma SGD dapat diamati pada grafik Gambar. 12, di mana akurasi pelatihan hanya mampu menyentuh angka 88%. Hasil tersebut menunjukkan ketertinggalan yang cukup signifikan apabila dikomparasikan dengan capaian optimizer Adam dan Adamax. Siklus pelatihan terhenti pada epoch ke-89 karena early stopping, dengan total waktu komputasi pelatihan selama 15 jam 1 menit 10 detik. Hal ini mengkonfirmasi bahwa penggunaan optimizer SGD pada arsitektur MobileNetV2 menuntut durasi pemrosesan yang paling lama dibandingkan kedua optimizer lainnya.



Gambar. 12 Akurasi Training MobileNetV2 dengan SGD Optimizer

Berdasarkan hasil pengujian klasifikasi pada Gambar. 13 yang dievaluasi melalui confusion matrix menggunakan data testing, performa arsitektur MobileNetV2 dengan optimizer SGD terbukti berada di bawah capaian optimizer Adam dan Adamax. Model ini mencatatkan tingkat keberhasilan sebesar 90% pada metrik precision, serta skor 88% yang terdistribusi merata pada indikator akurasi, recall, dan F1-score. Secara keseluruhan, perolehan metrik tersebut menunjukkan ketertinggalan kinerja yang cukup signifikan apabila

dikomparasikan dengan kedua algoritma optimasi sebelumnya. Nilai recall dan F1-score yang lebih rendah mengindikasikan bahwa model masih mengalami kesulitan dalam mengidentifikasi seluruh sampel secara konsisten, sehingga menghasilkan lebih banyak kesalahan klasifikasi. Hasil ini menunjukkan bahwa penggunaan SGD pada konfigurasi penelitian ini belum mampu memberikan kemampuan generalisasi yang sebaik Adam maupun Adamax dalam tugas klasifikasi citra wajah asli dan wajah hasil generatif AI.

	precision	recall	f1-score	support
0	0.816	0.992	0.895	12137
1	0.990	0.776	0.870	12147
accuracy			0.884	24284
macro avg	0.903	0.884	0.883	24284
weighted avg	0.903	0.884	0.883	24284

Gambar. 13 Hasil Evaluasi Confusion Metrik pada Model dengan SGD Optimizer

D. Implementasi Aplikasi Mobile

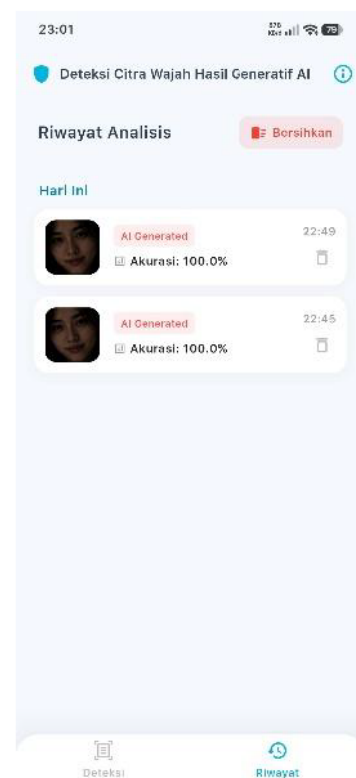


Gambar. 14 Tampilan Aplikasi Halaman Deteksi

Pada Gambar. 14 terdapat Antarmuka sistem dirancang dengan tata letak yang modern, minimalis, dan mudah dioperasikan. Fungsionalitas aplikasi dibagi menjadi dua halaman utama yang dapat diakses melalui panel navigasi bawah, yaitu halaman Deteksi dan halaman Riwayat. Pemisahan fungsi ini bertujuan untuk memudahkan pengguna dalam melakukan proses analisis sekaligus meninjau kembali hasil deteksi yang telah dilakukan sebelumnya.

Sedangkan pada Gambar. 15 terdapat halaman antarmuka riwayat yang berfungsi untuk menyimpan dan menampilkan seluruh hasil prediksi yang telah

dilakukan oleh pengguna. Data riwayat prediksi direpresentasikan secara terstruktur dan dikelompokkan berdasarkan tanggal analisis, seperti hari ini atau kemarin, sehingga pengguna dapat dengan mudah menelusuri hasil deteksi yang telah dilakukan sebelumnya. Setiap entri riwayat memuat informasi hasil klasifikasi beserta gambar yang dianalisis, sehingga pengguna dapat melakukan peninjauan ulang terhadap hasil prediksi tanpa harus mengulangi proses deteksi. Penyajian data dalam bentuk daftar yang terorganisasi membantu meningkatkan kemudahan navigasi, terutama ketika jumlah riwayat analisis semakin banyak. Pengelompokan berdasarkan waktu juga memudahkan pengguna dalam membedakan hasil deteksi terbaru dengan hasil deteksi yang lebih lama. Selain berfungsi sebagai dokumentasi hasil analisis, halaman ini juga dapat dimanfaatkan untuk membandingkan hasil prediksi dari beberapa gambar yang berbeda. Untuk menjaga efisiensi penyimpanan pada perangkat, pengguna diberikan kontrol penuh untuk menghapus data riwayat, baik secara individual maupun secara keseluruhan sesuai dengan kebutuhan. Dengan adanya fitur pengelolaan riwayat tersebut, aplikasi tidak hanya berfungsi sebagai alat deteksi, tetapi juga memberikan kemudahan dalam mengelola dan mengevaluasi hasil analisis yang telah dilakukan sebelumnya.



Gambar. 15 Tampilan Aplikasi Halaman Riwayat

E. Pengujian Aplikasi Mobile

Pengujian ini bertujuan untuk memastikan bahwa seluruh fitur aplikasi berjalan sesuai dengan spesifikasi yang diharapkan dan model dapat melakukan klasifikasi gambar dengan akurat pada lingkungan perangkat mobile. Pengujian dibagi menjadi dua tahap, yaitu pengujian fungsionalitas (black-box testing) dan pengujian analisis gambar.

TABEL 6
BLACKBOX TESTING

No	Skenario Pengujian	Hasil yang Diharapkan	Kesimpulan
1	Menekan tombol "Pilih dari Galeri" pada halaman Deteksi.	Sistem berhasil membuka galeri perangkat dan menampilkan gambar yang dipilih pada area pratinjau (<i>preview</i>).	Valid
2	Menekan tombol "ANALISIS SEKARANG" tanpa memilih gambar.	Sistem menahan proses dan menampilkan pesan peringatan (<i>SnackBar</i>) "Silakan pilih gambar terlebih dahulu!".	Valid
3	Menekan tombol "ANALISIS SEKARANG" dengan memilih gambar tanpa wajah manusia.	Sistem menahan proses dan menampilkan pesan peringatan (<i>SnackBar</i>) "Gambar wajah manusia tidak ditemukan".	Valid
4	Menekan tombol "ANALISIS SEKARANG" setelah gambar dipilih.	Sistem memproses gambar, mengubah teks tombol menjadi "SUDAH DIANALISIS", dan memunculkan rencana hasil prediksi beserta akurasi.	Valid
5	Membuka halaman "Riwayat" melalui panel navigasi bawah.	Sistem berpindah halaman dan menampilkan daftar riwayat deteksi yang dikelompokkan berdasarkan tanggal secara terstruktur.	Valid
6	Menekan ikon hapus (tempat sampah) pada salah satu	Sistem menampilkan dialog konfirmasi,	Valid

No	Skenario Pengujian	Hasil yang Diharapkan	Kesimpulan
	item riwayat.	dan jika disetujui, item tersebut berhasil dihapus dari antarmuka	
7	Menekan tombol "Bersihkan" (Hapus Semua) pada halaman Riwayat.	Sistem menampilkan dialog konfirmasi, dan jika disetujui, seluruh data riwayat terhapus dan menampilkan pesan "Belum ada riwayat".	Valid

Pada pengujian black-box testing semua fitur atau sistem aplikasi valid, yang artinya berjalan lancar tanpa adanya bug ataupun crash. Sedangkan pengujian analisis gambar, terbukti mampu mengeksekusi inferensi model pada perangkat mobile. Aplikasi cenderung mampu mengklasifikasikan kelas dengan tepat pada berbagai kondisi visual, termasuk pada gambar dengan pencahayaan gelap maupun gambar dengan wajah yang hanya terlihat sebagian. Namun terdapat 3 dari 4 kesalahan merupakan gambar dengan jarak wajah jauh. Pengujian analisis gambar pada penelitian ini menggunakan empat skema pengujian yaitu sumber gambar, pencahayaan, posisi wajah, dan jarak wajah. Dari masing masing skema pengujian tersebut, peneliti juga melihat kondisi uji. Berikut pada tabel 6 merupakan skema pengujian yang digunakan.

TABEL 6
SKEMA PENGUJIAN ANALISIS

No	Skema Pengujia	Kondisi Uji
1	Sumber gambar	ChatGPT
		Gemini
		Canva AI
		Kamera smartphone
2	Pencahayaan	Jelas
		Redup
3	Posisi wajah	Tegak lurus
		Miring
4	Jarak wajah	Dekat
		Jauh

IV. KESIMPULAN

Berdasarkan hasil penelitian, implementasi, dan pengujian yang telah dilakukan pada sistem deteksi gambar wajah manusia hasil generatif AI, dapat disimpulkan bahwa proses pelatihan arsitektur MobileNetV2 berhasil dilakukan dengan memproses 161.887 gambar wajah yang telah dipotong menggunakan algoritma MediaPipe BlazeFace, diubah ke dimensi 224x224 piksel, dinormalisasi, serta dibagi menggunakan

rasio dataset 70:15:15. Melalui komparasi tiga optimizer (Adam, Adamax, dan SGD) dengan learning rate 5e-6 dan penerapan early stopping, eksperimen menunjukkan bahwa optimizer Adam merupakan algoritma yang paling optimal dalam mencapai titik konvergensi. Keunggulan tersebut dibuktikan dengan tingkat akurasi training dan validation yang masing-masing mencapai 99%, serta akurasi testing sebesar 99% dengan capaian metrik rata-rata Precision, Recall, dan F1-Score yang juga menyentuh angka 99%, menandakan performa klasifikasi yang sangat baik.

Selanjutnya, model terbaik berformat asli (.h5) ini sukses dikonversi menjadi TensorFlow Lite (.tflite) dan diintegrasikan ke dalam aplikasi mobile berbasis Flutter. Dengan memanfaatkan SQLite sebagai penyimpanan lokal, proses inferensi dapat berjalan secara langsung pada perangkat (on-device) tanpa membutuhkan koneksi server. Hasil pengujian akhir menunjukkan bahwa aplikasi ini sangat fungsional karena mampu mengklasifikasikan 35 gambar dengan benar dari total 39 sampel uji, meskipun masih terdapat 4 kesalahan prediksi di mana 3 di antaranya disebabkan oleh kondisi gambar wajah yang berjarak jauh.

Berdasarkan proses dan hasil penelitian yang telah dijabarkan, terdapat beberapa saran yang dapat dipertimbangkan untuk pengembangan sistem pada penelitian selanjutnya. Penelitian berikutnya disarankan untuk menambahkan variasi dataset atau menerapkan teknik augmentasi khusus yang memfokuskan pada gambar wajah dengan kondisi jarak jauh atau beresolusi rendah guna menutupi kelemahan deteksi model pada kondisi visual tersebut

REFERENSI

- [1] S. J. Nightingale and H. Farid, "AI-synthesized faces are indistinguishable from real faces and more trustworthy," *Proc. Natl. Acad. Sci. U. S. A.*, vol. 119, no. 8, pp. 2–4, 2022, doi: 10.1073/pnas.2120481119.
- [2] T. Karras and E. Härkönen, "Alias-Free Generative Adversarial Networks," no. NeurIPS, 2021.
- [3] T. Karras and T. Aila, "Analyzing and Improving the Image Quality of StyleGAN".
- [4] R. Rombach, A. Blattmann, D. Lorenz, and P. Esser, "High-Resolution Image Synthesis with Latent Diffusion Models," pp. 10684–10695.
- [5] R. Tolosana, R. Vera-rodriguez, J. Fierrez, A. Morales, and J. Ortega-garcia, "DeepFakes and Beyond : A Survey of Face Manipulation and Fake Detection," pp. 1–23, 2018.
- [6] S. Wang, O. Wang, U. C. Berkeley, R. Zhang, A. Owens, and A. A. Efros, "CNN-generated images are surprisingly easy to spot ... for now," 2019.
- [7] E. Igba, H. Salam Olarinoye, V. Ezech Nwakaego, D. Batur Sehemba, Y. Shade Oluhaiyero, and N. Okika, "Synthetic Data Generation Using Generative AI to Combat Identity Fraud and Enhance Global Financial Cybersecurity Frameworks," *Int. J. Sci. Res. Mod. Technol.*, vol. 4, no. 2, pp. 1–19, 2025.
- [8] Y. Mirsky and W. Lee, "The Creation and Detection of Deepfakes," *ACM Comput. Surv.*, vol. 54, no. 1, 2021, doi: 10.1145/3425780.
- [9] Y. Luo, "Generalizing Face Forgery Detection with High-frequency Features".
- [10] 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.
- [11] K. Sahithya, J. Keerthana, K. S. Joshi, and R. K. Basha, "Efficient Deepfake Image Detection Model Based on MobileNetV2," *Int. J. Multidiscip. Res. Growth Eval.*, vol. 6,

- no. 2, pp. 650–653, 2025, doi: 10.54660/ijmrge.2025.6.2.650-653.
- [12] E. Architecture, "G-Tech : Jurnal Teknologi Terapan," vol. 9, no. 4, pp. 1829–1838, 2025.