

TUGAS AKHIR

**DETEKSI GAMBAR BUATAN AI MENGGUNAKAN
TRANSFER LEARNING MOBILENETV2 DAN
TENSORFLOW LITE PADA APLIKASI ANDROID**



SYAFWAL GUNARDI

213510057

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS ISLAM RIAU

PEKANBARU

2025

HALAMAN PENGESAHAN TUGAS AKHIR

Nama : Syafwal Gunardi
NPM : 213510057
Kelompok Keahlian : Mobile Based Information System & Application
Program Studi : Teknik Informatika
Jenjang Pendidikan : Strata Satu (S1)
Judul TA : Deteksi Gambar Buatan Ai Menggunakan Transfer Learning
MobileNetV2 dan TensorFlow Lite pada Aplikasi Android.

Format sistematika dan pembahasan materi pada masing-masing bab dan sub bab dalam tugas akhir ini telah dipelajari dan dinilai relatif telah memenuhi ketentuan-ketentuan dan kriteria- kriteria dalam metode penelitian ilmiah. Oleh karena itu tugas akhir ini dinilai layak dapat disetujui untuk disidangkan dalam ujian **Seminar Tugas Akhir**.

Pekanbaru, 7 November 2025

Di sahkan oleh :

Penguji I



Anggi Hanafiah, S.Kom., M.Kom

Penguji II



Ana Yulianti, S.T., M.Kom.

Ketua Program Studi
Teknik Informatika



Dr Ir DES SURYANI, M.Sc
NIDN : 1026126801

Dosen Pembimbing



Octadino Haryadi, S.Kom., M.Kom
NIDN: 1031109201

HALAMAN PENGESAHAN
DEWAN PENGUJI TUGAS AKHIR

Nama : Syafwal Gunardi
NPM : 213510057
Kelompok Keahlian : Mobile Application
Program Studi : Teknik Informatika
Jenjang Pendidikan : Strata Satu (S1)
Judul TA : Deteksi Gambar Buatan AI Menggunakan Transfer
Learning MobilenetV2 dan Tensorflow Lite Pada Aplikasi
Android

Tugas Akhir ini secara keseluruhan dinilai telah memenuhi ketentuan-ketentuan dan kaidah-kaidah dalam penulisan penelitian ilmiah serta telah diuji dan dapat dipertahankan dihadapan dewan penguji. Oleh karena itu, Tim Penguji Ujian Tugas Akhir Fakultas Teknik Universitas Islam Riau menyatakan bahwa mahasiswa yang bersangkutan dinyatakan **Telah Lulus Mengikuti Ujian Tugas Akhir Pada Tanggal 08 Januari 2026** dan disetujui serta diterima untuk memenuhi salah satu syarat guna memperoleh gelar Sarjana Strata Satu Bidang Ilmu Teknik Informatika.

Pekanbaru, 08 Januari 2026

Dewan Penguji

- | | | |
|---------------|----------------------------------|---|
| 1. Pembimbing | : Octadino Haryadi, S.Kom, M.Kom | () |
| 2. Penguji 1 | : Anggi Hanafiah, S.Kom, M.Kom | () |
| 3. Penguji 2 | : Ana Yulianti, S.T, M.Kom | () |

Disahkan Oleh :

Ketua Program Studi
Teknik Informatika



Dr. Ir. Des Suryani, M. Sc
1026126801

PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa tugas akhir ini merupakan karya saya sendiri dan semua sumber yang tercantum didalamnya baik yang dikutip maupun dirujuk telah saya nyatakan dengan benar sesuai ketentuan. Jika terdapat unsur penipuan atau pemalsuan data maka saya bersedia dicabut gelar yang saya peroleh.

Pekanbaru, 4 Maret 2025

Penulis,



Syafwal Gunardi
213510057

KATA PENGANTAR

Puji dan syukur penulis panjatkan ke hadirat Allah SWT atas segala rahmat, hidayah, dan karunia-Nya sehingga penulis dapat menyelesaikan tugas akhir yang berjudul **“Deteksi Gambar Buatan AI Menggunakan *Transfer Learning* MobileNetV2 dan *TensorFlow Lite* pada Aplikasi Android”** ini dengan baik. Tugas akhir ini disusun sebagai salah satu syarat untuk menyelesaikan studi pada Program Studi Teknik Informatika, Fakultas Teknik, Universitas Islam Riau.

Penulis menyadari sepenuhnya bahwa penyusunan tugas akhir ini tidak akan terwujud tanpa adanya bantuan, dukungan, serta bimbingan dari berbagai pihak. Oleh karena itu, dengan segala kerendahan hati, penulis ingin menyampaikan rasa terima kasih yang tulus kepada:

1. Allah SWT, yang telah melimpahkan kemudahan, kesehatan, dan kelancaran dalam setiap langkah penulis selama menyelesaikan tugas akhir ini.
2. Kedua orang tua dan seluruh keluarga tercinta, yang tidak henti-hentinya memberikan dukungan moril, materil, serta doa yang tiada putus kepada penulis.
3. Bapak Octadino Haryadi, S.Kom., M.IP., M.Kom., selaku dosen pembimbing yang telah dengan sabar memberikan bimbingan, arahan, serta motivasi yang sangat berharga dalam penyusunan tugas akhir ini.
4. Ibuk Dr. Ir Des Suryani, M.Sc., selaku Ketua Program Studi Teknik Informatika yang telah memberikan kemudahan dalam urusan akademik.
5. Seluruh dosen dan staf pengajar Program Studi Teknik Informatika Universitas Islam Riau, yang telah memberikan ilmu dan wawasan yang sangat bermanfaat selama masa studi.
6. Program Studi Independent Bangkit *Academy*, yang telah memberikan kesempatan belajar, wawasan industri, dan pengetahuan mendalam di bidang Android (*Kotlin*).

7. Himpunan Mahasiswa Teknik Informatika (HIMATIF), yang telah menjadi wadah untuk berkembang dan menjalin relasi selama masa perkuliahan.
8. Teman-teman seperjuangan, yang selalu memberikan semangat, inspirasi, dan doa dalam penyelesaian tugas akhir ini.
9. Semua pihak yang telah berperan membantu dalam penulisan dan penyelesaian tugas akhir ini yang tidak dapat penulis sebutkan satu per satu.

Penulis menyadari bahwa tugas akhir ini masih jauh dari kata sempurna. Oleh karena itu, segala bentuk kritik dan saran yang membangun sangat penulis harapkan untuk perbaikan di masa mendatang. Semoga tugas akhir ini dapat memberikan manfaat bagi pengembangan ilmu pengetahuan serta menjadi kontribusi dalam bidang pengembangan *software*, *computer vision* dan keamanan digital.

Pekanbaru, 12 Juni 2025

Penulis,



Syafwal Gunardi
NPM. 213510057

DAFTAR ISI

HALAMAN PENGESAHAN TUGAS AKHIR

HALAMAN PENGESAHAN DEWAN PENGUJI TUGAS AKHIR

PERNYATAAN KEASLIAN TUGAS AKHIR

KATA PENGANTAR..... i

DAFTAR ISI..... iii

DAFTAR GAMBAR vi

DAFTAR TABEL viii

ABSTRAK ix

ABSTRACT x

BAB I PENDAHULUAN 1

1.1 Latar Belakang 1

1.2 Identifikasi Masalah 4

1.3 Rumusan Masalah 5

1.4 Batasan Masalah 5

1.5 Tujuan Penelitian 6

1.6 Manfaat Penelitian 7

BAB II LANDASAN TEORI 8

2.1 Tinjauan Pustaka 8

2.2 Dasar Teori 12

2.2.1 Kecerdasan Buatan (Artificial Intelligence) 12

2.2.2 Gambar Buatan AI (AI-Generated Images) 12

2.2.3 Machine Learning 14

2.2.4 Deep Learning 16

2.2.5 Transfer Learning 17

2.2.6 Convolutional Neural Network (CNN) 17

2.2.7 MobileNetV2 21

2.2.8	TensorFlow Lite	22
2.2.9	Dataset.....	23
2.2.10	Data Preprocessing.....	23
2.2.11	Data Cleaning.....	24
2.2.12	Augmentasi Data.....	24
2.2.13	Confusion Matrix	25
2.3	Kerangka Pemikiran.....	27
BAB III METODOLOGI PENELITIAN		29
3.1	Jenis Penelitian	29
3.2	Objek Penelitian.....	29
3.3	Alat dan Bahan Penelitian.....	30
3.3.1	Spesifikasi Kebutuhan Hardware.....	30
3.3.2	Spesifikasi Kebutuhan Software dan Data.....	31
3.4	Alur Penelitian	33
3.4.1	Pengumpulan Data	33
3.4.2	Metode Analisis Data	38
3.5	Pra–Pemrosesan Data	38
3.5.1	Pra-pemrosesan dan Pemisahan Data (Split Data)	39
3.5.2	Augmentasi Data.....	41
3.5	Perancangan Arsitektur dan Pelatihan Model	43
3.7	Konversi Model ke TensorFlow Lite (TFLite).....	45
BAB IV HASIL DAN PEMBAHASAN.....		46
4.1	Antarmuka Perangkat Lunak.....	46
4.1.1	Alur Logis Aplikasi.....	46
4.2	Hasil Penelitian	53
4.2.1	Hasil Evaluasi Kinerja Model AI	53
4.2.2	Hasil Analisis Nilai Ambang Batas (<i>Threshold Analysis</i>).....	56
4.2.3	Hasil Kinerja Implementasi On-Device	61
4.2.4	Hasil Pengujian Fungsional (<i>Black Box Testing</i>)	63
4.2.5	Hasil Pengujian User Acceptance Testing (UAT)	65
4.3	Pembahasan.....	73
BAB V KESIMPULAN DAN SARAN		74

5.1	Kesimpulan	74
5.2	Saran.....	76
DAFTAR PUSTAKA		78



DAFTAR GAMBAR

Gambar 2. 1 Gambar Wajah dari dataset Kaggle 140k Real and Fake Face	14
Gambar 2. 2 Logo TensorFlow Lite	22
Gambar 2. 3 Kerangka Pemikiran.	28
Gambar 3. 1 Dataset yang digunakan dari sumber Open Source (Kaggle)	34
Gambar 3. 2 Dataset yang digunakan dari sumber Open Source (Kaggle)	35
Gambar 3. 3 Dataset Foto Real yang diambil dari kamera Smartphone	36
Gambar 3. 4 Flowchart Penelitian	38
Gambar 3. 5 Implementasi Kode untuk Pembagian Dataset (Split Folders) Menjadi Data Latih, Validasi, dan Uji.....	40
Gambar 3. 6 Implementasi Kode Augmentasi Data Menggunakan ImageDataGenerator.....	42
Gambar 3. 7 Rancangan Arsitektur Model Transfer Learning Menggunakan MobileNetV2.....	43
Gambar 3. 8 Implementasi Kode Penambahan Head Model Kustom (GlobalAveragePooling, Dense, Dropout).	44
Gambar 3. 9 Implementasi Kode Penggabungan Model dan Kompilasi dengan Learning Rate Rendah.....	44
Gambar 3. 10 Implementasi Kode Konversi Model Keras (.h5) Menjadi TensorFlow Lite (.tflite).	45
Gambar 4. 1 Tampilan Halaman Onboarding Aplikasi Fakelyze.....	47

Gambar 4. 2 Tampilan Halaman Login dan Register untuk Autentikasi Pengguna.	48
Gambar 4. 3 Tampilan Halaman Utama (Home) dengan Opsi Input Kamera dan Galeri.....	49
Gambar 4. 4 Tampilan Halaman Hasil Deteksi untuk Kategori Gambar Asli dan Gambar Buatan AI.....	50
Gambar 4. 5 Tampilan Halaman Riwayat (History) Penyimpanan Hasil Deteksi.	51
Gambar 4. 6 Tampilan Halaman Profil dan Statistik Aktivitas Pengguna.....	52
Gambar 4. 7 Grafik Pergerakan Akurasi dan Loss Selama 16 Epoch.	53
Gambar 4. 8 Classification Report (Precision, Recall, F1-score).....	55
Gambar 4. 9 Visualisasi Confusion Matrix Hasil Prediksi Model pada Data Uji	55
Gambar 4. 10 Hasil Deteksi Objek Asli dengan Variasi Skor Keyakinan (78%, 67%, 58%).....	59
Gambar 4. 11 Tantangan Deteksi: Kesalahan Klasifikasi (False Positive) pada Objek Nyata.	60
Gambar 4. 12 Rincian Properti File Model TensorFlow Lite (.tflite) Hasil Konversi.	61
Gambar 4. 13 Hasil Log Pengujian Kecepatan Inferensi (Timing) pada Perangkat Android.	62

DAFTAR TABEL

Tabel 2. 1 Perbandingan Tinjauan Pustaka.....	10
Tabel 3. 1 Spesifikasi Kebutuhan <i>Hardware</i>	30
Tabel 3. 2 Spesifikasi Kebutuhan Software dan Data	31
Tabel 3. 3 Jumlah Pembagian Dataset	37
Tabel 4. 1 Distribusi Rata-Rata Skor Keyakinan (Confidence Score)	57
Tabel 4. 2 Hasil Pengujian Onboarding & Autentikasi	63
Tabel 4. 3 Hasil Pengujian Deteksi Gambar.....	63
Tabel 4. 4 Hasil Pengujian Hasil dan Riwayat	64
Tabel 4. 5 Hasil Pengujian Profile	64
Tabel 4. 6 Bobot Jawaban.....	65
Tabel 4. 7 Hasil Jawaban Kuesioner.....	66
Tabel 4. 8 Jumlah Respon Kategori.....	70
Tabel 4. 9 Nilai Interval.....	72

DETEKSI GAMBAR BUATAN AI MENGGUNAKAN TRANSFER LEARNING MOBILENETV2 DAN TENSORFLOW LITE PADA APLIKASI ANDROID

SYAFWAL GUNARDI

Program Studi Teknik Informatika, Fakultas Teknik, Universitas Islam Riau

Email: syafwalgunardi@student.uir.ac.id

ABSTRAK

Perkembangan pesat teknologi kecerdasan buatan, khususnya *Generative Adversarial Networks* (GAN), telah memungkinkan pembuatan citra sintetis yang sangat realistis, sehingga sulit dibedakan dari gambar asli. Fenomena ini memicu kekhawatiran terkait penyebaran disinformasi dan hoaks visual. Sementara itu, model deteksi *Deep Learning* konvensional umumnya membutuhkan komputasi tinggi yang sulit diakses oleh perangkat seluler. Penelitian ini bertujuan untuk mengembangkan sistem deteksi gambar buatan AI yang efisien dan akurat pada platform Android menggunakan arsitektur *Convolutional Neural Network* (CNN) MobileNetV2 dengan teknik *Transfer Learning*. Model dilatih menggunakan dataset gabungan sebanyak 4.000 citra yang terdiri dari wajah asli dan buatan (StyleGAN) serta karya seni digital. Untuk optimalisasi kinerja pada perangkat *mobile* (*on-device*), model dikonversi menggunakan kerangka kerja TensorFlow Lite. Hasil pengujian menunjukkan bahwa model mampu mencapai akurasi sebesar 88% pada data uji. Implementasi pada perangkat Android (POCO F6) menunjukkan efisiensi tinggi dengan ukuran model akhir 9,07 MB dan kecepatan inferensi rata-rata 17 milidetik per gambar. Selain itu, pengujian *User Acceptance Testing* (UAT) memperoleh skor rata-rata 4,15, yang mengindikasikan bahwa aplikasi dapat diterima dengan baik oleh pengguna sebagai alat bantu verifikasi keaslian citra digital yang praktis.

Kata Kunci: *Artificial Intelligence, Android, Deep Learning, MobileNetV2, TensorFlow Lite, Transfer Learning.*

AI-GENERATED IMAGE DETECTION USING MOBILENETV2 TRANSFER LEARNING AND TENSORFLOW LITE ON ANDROID APPLICATION

SYAFWAL GUNARDI

*Informatics Engineering Study Program, Faculty of Engineering, Riau Islamic
University*

Email: syafwalgunardi@student.uir.ac.id

ABSTRACT

The rapid advancement of artificial intelligence technology, particularly Generative Adversarial Networks (GAN), has enabled the creation of highly realistic synthetic images that are difficult to distinguish from real ones. This phenomenon raises concerns regarding the spread of disinformation and visual hoaxes. Meanwhile, conventional Deep Learning detection models generally require high computational resources, making them inaccessible for mobile devices. This study aims to develop an efficient and accurate AI-generated image detection system on the Android platform using the MobileNetV2 Convolutional Neural Network (CNN) architecture with Transfer Learning techniques. The model was trained using a combined dataset of 4,000 images consisting of real and StyleGAN-generated faces, as well as digital art. For performance optimization on mobile devices (on-device), the model was converted using the TensorFlow Lite framework. Test results show that the model achieved an accuracy of 88% on the test data. Implementation on an Android device (POCO F6) demonstrated high efficiency with a final model size of 9.07 MB and an average inference speed of 17 milliseconds per image. Furthermore, the User Acceptance Testing (UAT) obtained an average score of 4.15, indicating that the application is well-received by users as a practical tool for verifying the authenticity of digital images.

Keywords: *Artificial Intelligence, Android, Deep Learning, MobileNetV2, TensorFlow Lite, Transfer Learning.*

BAB I

PENDAHULUAN

1.1 Latar Belakang

Evolusi teknologi Kecerdasan Buatan (AI), utamanya dalam domain *Computer Vision*, telah mengalami akselerasi radikal dalam satu dekade terakhir. Kehadiran arsitektur generatif modern, mulai dari *Generative Adversarial Networks* (GAN) hingga model difusi, telah memungkinkan sintesis citra digital dengan tingkat realisme yang presisi. Fenomena ini menciptakan dualitas dampak: di satu sisi mendorong efisiensi sektor industri kreatif, namun di sisi lain memicu kerentanan baru berupa disinformasi visual dan fabrikasi identitas digital yang sulit divalidasi secara manual.

Di satu sisi, teknologi AI generatif memberikan kemudahan dan efisiensi dalam bidang industri kreatif, hiburan, hingga arsitektur. Namun, di sisi lain, keberadaan gambar buatan AI juga menimbulkan potensi penyalahgunaan, seperti penyebaran informasi palsu (*hoaks* visual), manipulasi identitas (*deepfake*), serta pelanggaran hak cipta. Dengan semakin banyaknya gambar buatan AI yang beredar luas di media sosial dan internet, kemampuan masyarakat umum untuk mengenali dan membedakan gambar buatan dari gambar nyata menjadi semakin penting.

Di tengah banjirnya konten sintetis di media sosial, masyarakat membutuhkan instrumen verifikasi yang tidak hanya akurat tetapi juga mudah diakses. Sayangnya, mayoritas algoritma deteksi *Deep Learning* konvensional menuntut beban komputasi masif yang tidak kompatibel dengan perangkat seluler. Oleh karena itu, penelitian ini mengusulkan strategi deteksi yang mengadaptasi efisiensi

arsitektur *MobileNetV2* melalui teknik *Transfer Learning*, memungkinkan proses verifikasi keaslian citra berjalan langsung (*on-device*) pada ekosistem Android dengan latensi minimal.

Dalam bidang kecerdasan buatan, pendekatan *Deep Learning* telah banyak digunakan untuk klasifikasi dan deteksi objek pada gambar. Namun, model *Deep Learning* pada umumnya membutuhkan sumber daya komputasi yang besar, sehingga tidak ideal untuk diterapkan secara langsung pada perangkat *Mobile*. Untuk mengatasi hal ini, digunakanlah metode *Transfer Learning*, yaitu pendekatan pembelajaran yang memanfaatkan model *pralatih* (*pre-trained model*) agar proses pelatihan lebih ringan dan hasil lebih optimal. Salah satu model yang banyak digunakan untuk keperluan ini adalah *MobileNetV2*, yaitu arsitektur CNN (*Convolutional Neural Network*) ringan yang dirancang untuk efisiensi tinggi pada perangkat *Mobile* dengan tetap mempertahankan performa yang baik.

Gambar buatan AI yang digunakan dalam penelitian ini bersumber dari gabungan beberapa *dataset* untuk memastikan keragaman data. Dataset utama mencakup 140k *Real and Fake Faces*, yang berisi gambar wajah realistis dan gambar wajah buatan yang dihasilkan oleh model *Generative Adversarial Networks* (GAN), khususnya arsitektur *StyleGAN*. Selain itu, digunakan pula *dataset AI and Human Art Classification* yang memperluas variasi data dengan menyertakan karya seni yang dibuat oleh manusia dan yang dihasilkan oleh AI, sehingga model tidak hanya belajar dari citra *fotorealistis*. Untuk memperkaya koleksi gambar asli, penelitian ini juga menyertakan 196 foto orisinal yang diambil langsung menggunakan kamera ponsel.

Dengan menggunakan kombinasi *dataset* tersebut, model yang dikembangkan akan dioptimalkan untuk mendeteksi berbagai jenis artefak digital yang khas dari model-model generatif yang ada di dalam *dataset* tersebut. Oleh karena itu, pengujian tidak akan difokuskan pada gambar yang dihasilkan oleh model generatif yang lebih baru (misalnya Midjourney atau DALL-E 3) untuk menghindari potensi penurunan akurasi akibat pergeseran domain (*domain shift*).

Agar model *Deep Learning* tersebut dapat berjalan secara optimal di Android, maka diperlukan proses konversi ke dalam format yang lebih ringan menggunakan *TensorFlow Lite (TFLite)*. *TensorFlow Lite* memungkinkan model *Deep Learning* dijalankan di perangkat Mobile dengan penggunaan memori dan energi yang lebih efisien.

Merespons tantangan tersebut, penelitian ini mengajukan pengembangan sistem deteksi gambar berbasis AI yang efisien dan mudah diakses melalui aplikasi Android. Pendekatan yang digunakan adalah *transfer learning* dengan menggunakan arsitektur *MobileNetV2*. Arsitektur ini dikenal karena desainnya yang ringan dan efisien, menjadikannya sangat cocok untuk aplikasi pada perangkat mobile. Dengan memanfaatkan bobot pra-pelatihan dari *dataset* besar seperti *ImageNet*, *transfer learning* memungkinkan model untuk mengidentifikasi fitur penting dari gambar dengan membutuhkan lebih sedikit data pelatihan serta mempercepat proses pelatihan. Kemudian, model yang telah dilatih akan dikonversi menggunakan *TensorFlow Lite*, *framework* yang dirancang khusus untuk menjalankan inferensi model *machine learning* pada perangkat *edge*, termasuk *smartphone Android*. Implementasi ini diharapkan dapat menyediakan alat praktis

dan real-time bagi pengguna untuk memverifikasi keaslian gambar, sehingga turut berperan dalam mengurangi dampak negatif dari penyebaran gambar buatan AI.

Meskipun model generatif terkini seperti Midjourney dan DALL-E 3 menunjukkan tingkat realisme yang semakin tinggi, penelitian ini secara sadar memfokuskan pengujian pada gambar yang dihasilkan oleh arsitektur StyleGAN. Pilihan ini bersifat metodologis, di mana tujuan utamanya adalah untuk membuktikan kelayakan dan efisiensi arsitektur ringan (MobileNetV2 dan TensorFlow Lite) sebagai solusi deteksi *on-device* pada *dataset benchmark* yang terkontrol dan tervalidasi. Keberhasilan validasi pada fondasi ini akan menjadi landasan ilmiah yang kuat untuk pengembangan sistem di masa depan yang mampu menangani variasi model generatif yang lebih luas dan kompleks.

1.2 Identifikasi Masalah

Berdasarkan latar belakang di atas, didapatkan identifikasi masalah sebagai berikut :

1. Maraknya penyebaran gambar buatan AI yang sulit dibedakan dari gambar nyata.
2. Kurangnya aplikasi berbasis *Mobile* yang dapat mendeteksi gambar buatan AI secara efisien.
3. Tantangan dalam menerapkan model *Deep Learning* pada perangkat *mobile* yang memiliki keterbatasan memori dan daya komputasi

1.3 Rumusan Masalah

Berdasarkan latar belakang yang telah dipaparkan, didapatkan masalah penelitian ini adalah :

1. Bagaimana merancang dan melatih model klasifikasi gambar untuk membedakan gambar buatan *AI* dan gambar asli menggunakan arsitektur *MobileNetV2*?
2. Bagaimana cara mengonversi dan mengintegrasikan model tersebut ke dalam aplikasi Android menggunakan *TensorFlow Lite*?
3. Bagaimana performa model dan aplikasi dalam aspek *akurasi*, *efisiensi*, dan kemudahan penggunaan?

1.4 Batasan Masalah

Adapun batasan masalah dalam perancangan sistem ini sebagai berikut :

1. Gambar buatan AI yang digunakan sebagai data latih dan uji terbatas pada citra yang dihasilkan oleh arsitektur *Generative Adversarial Networks* (GAN), khususnya *StyleGAN*. Model ini dipilih secara metodologis karena merupakan salah satu arsitektur yang paling terdokumentasi, dan *dataset* publiknya (seperti *140k Real and Fake Faces*) berskala besar serta telah banyak digunakan sebagai *benchmark*, sehingga ideal untuk melatih dan mengevaluasi performa model deteksi secara adil dan dapat direproduksi.

Penelitian ini tidak akan melakukan pengujian pada gambar dari model generatif yang lebih baru seperti *Midjourney* atau *DALL-E 3* untuk menjaga fokus validasi arsitektur pada domain yang terkontrol. *Dataset* yang digunakan bersumber dari *dataset* terbuka (*Open-Source*) yang telah diklasifikasikan antara gambar asli dan gambar buatan.

2. Model *deep learning* yang digunakan terbatas pada *MobileNetV2* dengan *transfer learning*.
3. Aplikasi hanya dikembangkan untuk platform *Android* dan hanya diuji pada perangkat *android*.
4. Model hanya membedakan dua kelas: *Real* dan *Fake*.

1.5 Tujuan Penelitian

Adapun tujuan penelitian ini adalah :

1. Mengembangkan model deteksi gambar buatan AI menggunakan pendekatan *transfer learning* dengan *MobileNetV2*.
2. Mengimplementasikan model tersebut ke dalam aplikasi *Android* menggunakan *TensorFlow Lite*.
3. Menguji akurasi dan kinerja aplikasi dalam mendeteksi gambar buatan AI pada perangkat *mobile*.

1.6 Manfaat Penelitian

Manfaat dari penelitian ini adalah sebagai berikut :

1. Akademik

Memberikan kontribusi terhadap pengembangan teknologi deteksi konten *AI-generated* berbasis *deep learning*, *Mobile Computing* dan menjadi referensi bagi penelitian selanjutnya dalam bidang keamanan digital dan *computer vision* di platform *mobile*.

2. Praktis

Menyediakan alat bantu berupa aplikasi Android yang dapat membantu masyarakat umum dalam mengidentifikasi gambar buatan AI. Meningkatkan kesadaran terhadap potensi penyalahgunaan teknologi AI dalam pembuatan konten visual.

BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

Validitas penggunaan arsitektur *MobileNetV2* pada lingkungan perangkat bergerak telah dikonfirmasi oleh berbagai studi literatur. Dalam ranah klasifikasi objek *real-time*, arsitektur ini terbukti mampu memberikan performa tinggi dengan akurasi di atas 90%, sebagaimana ditunjukkan dalam pengembangan sistem deteksi objek umum (Komputasi et al., 2020) dan alat bantu pengenalan buah bagi penyandang disabilitas visual (Prima, 2023).

Penelitian oleh (Prima, 2023) dalam penelitiannya yang berjudul "*Rancang Bangun Sistem Pendeteksi Aneka Ragam Buah Menggunakan MobileNetV2*" mengembangkan aplikasi Android yang dapat mengenali berbagai jenis buah melalui kamera *smartphone*. Sistem ini dirancang khusus untuk membantu penyandang tunanetra. Model *MobileNetV2* dipilih karena efisien dan ringan, sehingga cocok untuk perangkat Android dengan spesifikasi terbatas.

Penelitian oleh (Muharram, 2021) dalam penelitiannya yang berjudul "*Implementasi Artificial Intelligence untuk Deteksi Masker Secara Realtime dengan TensorFlow dan SSD MobileNet Berbasis Python*" bertujuan untuk membangun sistem deteksi masker secara *real-time*. Model yang digunakan mampu mendeteksi kehadiran masker pada wajah dengan performa yang stabil, serta menunjukkan efektivitas pemanfaatan *TensorFlow* untuk aplikasi berbasis visi komputer.

Penelitian oleh (Aziz et al., 2024) dalam penelitiannya yang berjudul "*Pengujian Deteksi Objek Manusia Menggunakan Jetson Nano dengan Model SSD MobileNetV2*" mengevaluasi performa deteksi objek pada platform Jetson Nano. Penelitian ini menguji akurasi dalam berbagai kondisi cahaya dan jumlah objek, dengan hasil menunjukkan bahwa *MobileNetV2* mengalami penurunan akurasi pada kondisi lingkungan nyata yang kompleks.

Penelitian oleh (Mutianniza & Suwardoyo, 2023) dalam penelitiannya yang berjudul "*Aplikasi Kamera Cerdas Untuk Deteksi Kendaraan Menggunakan Library TensorFlow*" mengembangkan sistem pendeteksi kendaraan berbasis kamera yang terintegrasi dengan *TensorFlow*. Sistem ini dapat mengenali berbagai jenis kendaraan seperti mobil dan motor secara langsung di perangkat Android.

Penelitian oleh (Angeline & Kusniyati, 2024) dalam penelitiannya yang berjudul "Komparasi Performa VGG19, ResNet50, DenseNet121 dan MobileNetV2 Dalam Mendeteksi Gambar Deepfake" melakukan perbandingan performa beberapa arsitektur Convolutional Neural Network (CNN) untuk mendeteksi gambar *deepfake* (gambar buatan AI). Hasil penelitian menunjukkan bahwa arsitektur MobileNetV2 berhasil mencapai tingkat akurasi tertinggi sebesar 95,89% dan memiliki waktu komputasi tercepat dibandingkan model lainnya. Penelitian ini mengonfirmasi bahwa MobileNetV2 merupakan model yang sangat efektif dan efisien untuk tugas deteksi gambar buatan AI, yang memperkuat dasar pemilihan arsitektur ini untuk pengembangan aplikasi pada perangkat *mobile*.

Penelitian yang dilakukan oleh (Noprisson et al., 2024) dengan judul "MobileNet Performance Improvements for Deepfake Image Identification Using

Activation Function and Regularization" berfokus pada peningkatan performa model MobileNet untuk mengidentifikasi gambar *deepfake* melalui optimisasi menggunakan fungsi aktivasi dan teknik regularisasi. Dalam penelitiannya, penerapan regularisasi *Dropout* 0.5 pada model MobileNet yang menggunakan *transfer learning* berhasil meningkatkan akurasi deteksi hingga mencapai 96,39%. Studi ini membuktikan bahwa performa dasar dari MobileNetV2 dapat dioptimalkan lebih lanjut untuk menghasilkan sistem deteksi yang lebih akurat, memberikan landasan untuk penerapan teknik *fine-tuning* dalam pengembangan model.

Dari rangkuman seluruh tinjauan pustaka yang telah disusun, kita bisa mencermati perbandingan di antara tinjauan pustaka tersebut yang ditampilkan dalam Tabel 2.1.

Tabel 2. 1 Perbandingan Tinjauan Pustaka

Penulis	Tujuan	Metode Pengembangan	Fitur Utama	Hasil dan Relevansi
Aningtiyas et al. (2020)	Membangun aplikasi deteksi objek dengan MobileNetV2	TensorFlow Object Detection API, SSD MobileNetV2	Deteksi 5 objek (kamera, mouse, dsb)	Akurasi 93.02%. MobileNetV2 cocok untuk deteksi objek ringan berbasis Android
Ario Prima (2023)	Mendeteksi buah-buahan untuk tunanetra	Transfer Learning MobileNetV2, TensorFlow Lite	Deteksi real-time via kamera Android	Deteksi berhasil pada sebagian besar jenis buah,
Muharram & Suryadi (2021)	Deteksi masker wajah secara realtime	CNN, TensorFlow, Python	Deteksi masker dengan kamera	Model berjalan baik secara realtime. Relevan

				dengan klasifikasi wajah menggunakan MobileNet
Abdul Aziz et al. (2024)	Deteksi manusia dengan MobileNetV2 di Jetson Nano	SSD MobileNetV2, TensorFlow	Uji performa berbagai skenario cahaya	Akurasi menurun di pencahayaan buruk
Mutianniza & Suwardoyo (2022)	Mendeteksi kendaraan via kamera cerdas	TensorFlow, Android Studio	Deteksi berbagai jenis kendaraan	Sistem bekerja realtime. Mendukung konsep inferensi AI langsung di Android
(Angeline & Kusniyati, 2024)	Membandingkan model AI untuk deteksi <i>deepfake</i>	VGG19, ResNet50, DenseNet121, dan MobileNetV2	Komparasi akurasi dan kecepatan MobileNetV2 dengan model lain (VGG19, ResNet50, dll.)	MobileNetV2 terbukti paling akurat (95,89%) dan paling cepat, ideal untuk perangkat <i>mobile</i> .
Noprisson et al., 2024	Mengoptimalkan model MobileNet untuk deteksi <i>deepfake</i>	MobileNet, Regularisasi (Dropout), Transfer Learning	Penerapan teknik regularisasi untuk meningkatkan performa dan akurasi deteksi.	Akurasi meningkat hingga 96,39%, membuktikan teknik optimisasi sangat efektif untuk model deteksi.

Setelah melakukan tinjauan pustaka, dapat disimpulkan bahwa meskipun *MobileNetV2* dan *TensorFlow Lite* telah banyak digunakan dalam berbagai aplikasi *computer vision* di perangkat mobile, penelitian yang secara spesifik menargetkan deteksi gambar buatan AI untuk mitigasi disinformasi visual masih terbatas. Dari penelitian ini terletak pada penerapan *transfer learning MobileNetV2* secara khusus

untuk mengklasifikasikan gambar buatan AI (*real vs fake*) pada aplikasi Android yang praktis dan efisien. Kontribusi utamanya adalah menyediakan alat bantu *on-device* yang dapat diakses masyarakat luas secara *real-time* untuk memverifikasi keaslian gambar, yang merupakan jawaban langsung terhadap tantangan penyebaran *hoaks* visual.

2.2 Dasar Teori

2.2.1 Kecerdasan Buatan (Artificial Intelligence)

Kecerdasan buatan merupakan disiplin ilmu komputer yang memusatkan perhatian pada pengembangan sistem atau mesin yang bisa meniru kecakapan kognitif manusia, seperti belajar, berpikir, dan menyelesaikan masalah. Salah satu cabang yang mengalami banyak kemajuan dalam AI adalah *deep learning*, yang dimanfaatkan untuk mengenali pola dalam data yang lebih kompleks seperti gambar, suara, dan teks. Teknologi ini telah digunakan secara luas dalam berbagai penelitian untuk mendeteksi objek, ekspresi wajah, hingga menghasilkan citra buatan (Muharram & Suryadi, 2021; Aziz et al., 2024).

2.2.2 Gambar Buatan AI (AI-Generated Images)

Gambar buatan AI adalah citra digital yang dihasilkan oleh model generatif tanpa campur tangan manusia secara langsung. Model generatif yang umum digunakan meliputi *GAN* (*Generative Adversarial Networks*) dan *Diffusion Models*. Citra buatan ini biasanya memiliki kualitas sangat tinggi, sehingga sulit dibedakan dari gambar nyata. Karena kemiripannya dengan gambar asli, diperlukan pendekatan berbasis *deep learning* untuk mendeteksinya (Tiwari, 2021; Sivan, 2022).

Tantangan utama dalam mendeteksi gambar buatan AI terletak pada kemampuannya untuk menghasilkan citra yang sangat *fotorealistis*. Model generatif modern seperti GAN dan *Diffusion Models* secara aktif dilatih untuk meminimalkan *artefak digital* cacat halus pada gambar seperti pola tidak wajar pada mata, rambut, gigi, atau latar belakang yang pada generasi awal teknologi ini masih mudah dikenali. Seiring perkembangan model, artefak ini menjadi semakin subtil, sehingga memerlukan pendekatan *deep learning* yang memiliki kemampuan mendeteksi pola statistik rumit yang tak terlihat oleh mata manusia.

Untuk melatih dan menguji model deteksi, penelitian ini menggunakan gabungan beberapa *dataset* publik dan data *kustom*. Pendekatan ini bertujuan untuk membangun sebuah *dataset* yang kaya dan beragam. Salah satu komponen utamanya adalah *dataset* "140k Real and Fake Faces", yang dirancang khusus untuk tugas klasifikasi biner antara gambar wajah asli dan gambar buatan (*FAKE*). Gambar-gambar palsu dalam *dataset* ini dihasilkan menggunakan arsitektur model *Generative Adversarial Networks* (GAN), khususnya StyleGAN.

Selain itu, untuk memperluas kemampuan model di luar deteksi wajah, digunakan pula *dataset* "AI and Human Art Classification" yang berisi karya seni buatan manusia dan gambar yang dihasilkan oleh AI. Terakhir, untuk meningkatkan representasi gambar dunia nyata, *dataset* diperkaya dengan 196 foto orisinal yang diambil langsung menggunakan kamera ponsel. Kelengkapan dan keragaman dari gabungan *dataset* ini menjadikannya acuan yang kuat untuk melatih model deteksi berbasis *deep learning* yang lebih tangguh dan serbaguna.

Citra buatan AI dapat menyerupai gambar asli manusia dengan sangat realistis. Hal ini menyebabkan tantangan dalam membedakan antara gambar asli dan hasil rekayasa AI. Gambar di bawah ini merupakan contoh gambar dari *dataset* yang berisi pasangan gambar asli dan buatan.



Gambar 2. 1 Gambar Wajah dari *dataset* Kaggle 140k Real and Fake Face

2.2.3 Machine Learning

Konsep pembelajaran mesin pertama kali diperkenalkan oleh Arthur Samuel pada tahun 1959. Dia menggambarkan sebagai bidang studi yang memungkinkan mesin untuk memperoleh pengetahuan secara mandiri tanpa perlu diprogram secara khusus untuk tiap tugasnya. Seiring perkembangan waktu, definisi dasar ini telah berkembang. Penelitian terbaru, seperti yang ditekankan oleh Wei et al. (2023), mendefinisikan pembelajaran mesin sebagai upaya membangun model-model komputasi yang dapat mengekstraksi pengetahuan yang umum dari data berskala besar. Fokusnya kini telah bergeser dari penyelesaian masalah spesifik menuju penciptaan sistem yang cerdas, umum, dan mampu beradaptasi dengan berbagai domain aplikasi.

Secara metodologis, *machine learning* dapat diklasifikasikan ke dalam beberapa paradigma utama, antara lain:

1. Pembelajaran Terarah (*Supervised Learning*) Paradigma ini beroperasi menggunakan *dataset* yang setiap entrinya telah diberi anotasi atau kategori yang sesuai (data berlabel). Tujuannya adalah melatih model untuk mengenali hubungan kausal atau korelasional antara input dan output, sehingga ia dapat menghasilkan prediksi yang valid untuk data yang belum pernah ditemui. Salah satu implementasinya adalah pada sistem prediksi nilai properti.
2. Pembelajaran Tak Terarah (*Unsupervised Learning*) Beroperasi pada data tanpa label, tujuan utama pendekatan ini adalah untuk mengidentifikasi struktur, hierarki, atau pola yang tersembunyi secara inheren di dalam data. Algoritma secara mandiri mengelompokkan data berdasarkan kesamaan atribut, seperti dalam analisis segmentasi pasar untuk mengkategorikan konsumen.
3. Pembelajaran Penguatan (*Reinforcement Learning*) Di sini, sebuah 'agen' komputasi dilatih untuk membuat keputusan sekuensial dengan cara berinteraksi langsung dengan lingkungannya. Agen tersebut belajar dari konsekuensi tindakannya melalui mekanisme umpan balik berupa hadiah (*reward*) atau hukuman (*punishment*), yang bertujuan untuk menyusun strategi yang paling optimal. Paradigma ini sangat dominan dalam bidang robotika dan sistem otonom.
4. Pembelajaran Mandiri (*Self-Supervised Learning*) Sebagai motor penggerak utama di balik kecerdasan buatan modern, pendekatan ini memungkinkan

model untuk belajar dari data mentah dalam jumlah besar. Model belajar dengan menyelesaikan tugas-tugas buatan yang labelnya berasal dari data itu sendiri, misalnya, memprediksi bagian yang hilang dari sebuah kalimat. Kemampuan inilah yang menjadi fondasi bagi model bahasa skala besar.

Implementasi dari berbagai paradigma ini kini dapat ditemukan di hampir semua sektor teknologi terapan. Contohnya termasuk sistem AI generatif yang mampu memproduksi konten teks dan visual, platform rekomendasi yang sangat *terpersonalisasi*, perangkat lunak analisis citra medis, mekanisme deteksi anomali finansial, serta teknologi inti di balik kendaraan otonom. Kapabilitas canggih ini merupakan manifestasi dari keberhasilan model-model skala besar dalam menggeneralisasi informasi, sebuah fenomena yang dianalisis secara mendalam oleh Chen et al. (2024).

2.2.4 Deep Learning

Deep learning adalah bagian dari *machine learning* yang menerapkan jaringan saraf buatan dengan beragam lapisan, atau disebut jaringan saraf dalam. Dalam dunia visi komputer, *deep learning* diterapkan pada tugas seperti klasifikasi gambar, deteksi objek, dan segmentasi citra. Model populer seperti CNN, *MobileNet*, dan *ResNet* telah sukses diimplementasikan dalam berbagai penelitian untuk klasifikasi citra dan deteksi wajah (Aningtiyas et al., 2020; Prima, 2023; Muharram & Suryadi, 2021).

2.2.5 Transfer Learning

Transfer learning adalah teknik pelatihan model *deep learning* dengan memanfaatkan model yang telah dilatih sebelumnya pada dataset besar (seperti *ImageNet*), kemudian disesuaikan (*fine-tuned*) untuk menyelesaikan masalah yang berbeda. Dengan *transfer learning*, waktu pelatihan dapat dikurangi dan akurasi tetap tinggi, meskipun menggunakan *dataset* yang lebih kecil. Konsep ini sangat efektif pada penelitian pengembangan aplikasi *Mobile* yang terbatas dalam data pelatihan (Qin et al., 2023).

Dalam praktik penelitian ini, terdapat dua pendekatan utama dalam *transfer learning*. Pertama adalah *feature extraction*, di mana bobot dari model pra-latih sepenuhnya dibekukan (*frozen*) dan hanya lapisan klasifikasi baru yang dilatih. Kedua adalah *fine-tuning*, di mana sebagian lapisan atas dari model pra-latih ikut 'dicairkan' (*unfrozen*) dan dilatih kembali dengan *learning rate* yang sangat kecil.

2.2.6 Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) merupakan jenis jaringan saraf buatan yang khusus dikembangkan untuk menangani data dalam bentuk *grid*, seperti gambar. CNN berfungsi dengan mengaplikasikan filter (*kernel*) pada gambar input guna mengekstraksi elemen-elemen penting. CNN telah menjadi basis utama di berbagai aplikasi visi komputer, meliputi klasifikasi ekspresi wajah dan identifikasi objek. Dalam studi ini, CNN berperan sebagai "otak" dari sistem AI yang bertugas untuk "melihat" dan menganalisis piksel gambar untuk menemukan kejanggalan atau detail halus yang membedakan gambar buatan AI dari gambar asli. Struktur

CNN pada dasarnya terdiri dari sejumlah lapisan utama yang bekerja secara sinergis. (Muharram & Suryadi, 2021; Prima, 2023).

2.2.6.1 Lapisan Konvolusi (Convolution Layer)

Lapisan ini adalah inti dari CNN. Fungsinya adalah mengekstraksi fitur dari gambar input dengan melakukan operasi konvolusi antara matriks input (gambar) dengan filter atau *kernel*. *Kernel* ini adalah matriks kecil berisi bobot yang dapat dipelajari, yang digeser ke seluruh gambar untuk menghasilkan *feature map*. Setiap *feature map* merepresentasikan deteksi fitur spesifik, seperti tepi vertikal, sudut, atau tekstur tertentu. Proses ini memungkinkan model untuk mempertahankan hubungan spasial antar piksel dan mempelajari representasi visual secara hierarkis (Alzubaidi et al., 2021)

2.2.6.2 Lapisan Pooling (Pooling Layer)

Tujuan utama dari *pooling layer* adalah untuk mereduksi dimensi spasial dari *feature map* secara progresif. Proses ini, yang juga dikenal sebagai *downsampling*, biasanya diterapkan setelah lapisan konvolusi. Mekanismenya melibatkan penggeseran filter dengan ukuran tertentu melintasi *feature map*. Dua metode yang paling umum adalah:

1. *Max-Pooling*: Memilih nilai piksel tertinggi dari area yang dicakup oleh filter.
2. *Average-Pooling*: Menghitung nilai rata-rata dari semua piksel dalam area filter.

2.2.6.3 Fully-Connected Layer

Setelah sejumlah lapisan konvolusi dan pooling berhasil mengambil fitur-fitur tingkat lanjut, lapisan fully-connected (FC) berperan dalam melakukan klasifikasi. Lapisan ini menerima input dari peta fitur akhir (yang telah diratakan menjadi vektor satu dimensi) dan menghubungkan setiap neuron dari lapisan sebelumnya ke masing-masing neuron di lapisan saat ini. Lapisan FC inilah yang pada akhirnya akan mempelajari kombinasi fitur-fitur yang telah diekstrak untuk membuat keputusan akhir, seperti menentukan apakah sebuah gambar tergolong dalam kelas "Real" atau "Fake" (Alzubaidi et al., 2021).

2.2.6.4 Dropout Regularization

Dropout merupakan sebuah teknik regularisasi yang sangat ampuh digunakan untuk mencegah terjadinya *overfitting* pada jaringan saraf. Selama proses pelatihan, *dropout* secara acak menonaktifkan ("menjatuhkan") sebagian neuron pada lapisan tertentu pada setiap iterasi. Hal ini memaksa jaringan untuk tidak terlalu bergantung pada neuron-neuron tertentu dan mendorongnya untuk mempelajari representasi fitur yang lebih kuat dan terdistribusi. Dengan demikian, kemampuan generalisasi model pada data yang belum pernah dilihat sebelumnya dapat meningkat secara signifikan (Afnan et al., 2021).

2.2.6.5 Softmax Classifier

Fungsi aktivasi Softmax umumnya ditempatkan pada lapisan output dari model klasifikasi multi-kelas. Fungsi ini mengambil vektor nilai numerik mentah (*logits*) sebagai input dan mengubahnya menjadi distribusi probabilitas. Setiap nilai output akan berada di antara 0 dan 1, dan total dari semua nilai output adalah 1. Dalam konteks penelitian ini, Softmax akan menghasilkan probabilitas untuk setiap kelas (misalnya, 95% *Fake*, 5% *Real*), yang memungkinkan interpretasi tingkat kepercayaan model terhadap prediksinya (Sharma et al., 2020).

2.2.6.6 Cross-Entropy Loss Function

Untuk melatih sebuah model, dibutuhkan fungsi kerugian (loss function) yang berfungsi untuk menilai seberapa jauh prediksi model dari nilai sebenarnya. Pada tugas klasifikasi, Categorical Cross-Entropy sering digunakan bersama fungsi aktivasi Softmax. Fungsi ini digunakan untuk mengukur "jarak" antara distribusi probabilitas yang dihasilkan oleh model dengan distribusi probabilitas yang sesungguhnya (label asli). Tujuan dari pelatihan ini adalah untuk secara bertahap mengecilkan nilai kerugian melalui algoritma optimisasi, sehingga bobot-bobot dalam jaringan bisa diatur agar memberikan prediksi yang semakin tepat (Jadon, 2020).

2.2.7 MobileNetV2

Guna menjawab kebutuhan komputasi pada gawai bergerak (*mobile devices*), Google memperkenalkan MobileNetV2, sebuah arsitektur *Convolutional Neural Network* (CNN) yang diprioritaskan untuk efisiensi sumber daya. Arsitektur ini mengandalkan integrasi mekanisme *inverted residuals* dan *depthwise separable convolution*. Pendekatan tersebut memungkinkan model bekerja dengan bobot yang ringan dan latensi rendah, namun tetap mempertahankan tingkat akurasi yang kompetitif. ((Howard et al., 2017), (Tan et al., 2019) Model ini sangat cocok untuk aplikasi *Mobile* karena ukurannya kecil dan hemat sumber daya, serta banyak digunakan dalam penelitian deteksi objek dan klasifikasi gambar di Android (Aziz et al., 2024; Prima, 2023).

2.2.8 TensorFlow Lite

Untuk keperluan eksekusi model kecerdasan buatan pada perangkat *embedded* maupun seluler, Google menyediakan kerangka kerja khusus bernama *TensorFlow Lite* (TFLite). *Framework* ini berfungsi sebagai versi ringkas dari ekosistem *TensorFlow*, yang memungkinkan model hasil pelatihan dikonversi dan dikompresi agar mampu beroperasi secara optimal pada sistem operasi Android dengan konsumsi daya yang minimal. *TFLite* mendukung interpreter *runtime* yang cepat dan fleksibel, serta memungkinkan integrasi model dalam aplikasi Android menggunakan bahasa pemrograman seperti Kotlin atau Java (Tiwari, 2021; Lee et al., 2019; Xiong et al., 2020). Banyak penelitian berhasil mengimplementasikan klasifikasi gambar di Android menggunakan *TFLite* dengan hasil memuaskan (Sivan, 2022; Mutianniza & Suwardoyo, 2022).



Gambar 2. 2 Logo TensorFlow Lite

2.2.9 Dataset

Dataset merupakan elemen kunci dalam pengembangan model AI. *Dataset* besar seperti *ImageNet* digunakan dalam proses *pretraining*, sementara *dataset* khusus seperti citra buatan AI diperlukan dalam proses *fine-tuning* dan pengujian model. Kualitas dan keragaman *dataset* sangat mempengaruhi hasil akhir klasifikasi (Weirich, 2024; Prima, 2023), penjelasan lebih lanjut akan dijelaskan pada bab III.

2.2.10 Data Preprocessing

Tahap pra-pemrosesan (*preprocessing*) merupakan fase inisial yang krusial dalam alur kerja pengolahan citra. Tujuannya adalah untuk menyiapkan data mentah agar optimal untuk diproses oleh model (DLY et al., 2023). Fase ini mencakup serangkaian teknik untuk membersihkan, mentransformasi, dan menyusun ulang data, baik yang relevan maupun tidak, agar siap untuk tahap analisis atau pemodelan (Pujiono et al., 2024).

2.2.11 Data Cleaning

Untuk menjamin integritas dan kualitas dataset, dilakukan proses pembersihan data (*data cleaning*). Tahapan ini berfokus pada identifikasi dan penanganan data yang bermasalah, seperti informasi yang tidak lengkap, tidak akurat, tidak relevan, atau memiliki format yang salah. Tindakan yang dilakukan dapat berupa modifikasi, koreksi, atau penghapusan data tersebut dengan tujuan akhir menghasilkan dataset yang berkualitas tinggi untuk pelatihan model (Raharja et al., 2024).

2.2.12 Augmentasi Data

Augmentasi data adalah teknik yang esensial dalam pelatihan model, terutama ketika volume dataset terbatas. Tujuannya adalah untuk memperbanyak jumlah data secara artifisial melalui berbagai transformasi. Dengan dataset yang lebih besar dan bervariasi, model didorong untuk mempelajari fitur yang lebih *robust*, sehingga kemampuan generalisasinya meningkat dan kinerjanya menjadi lebih optimal (Auliyah Agustina, Febi Yanto, Elvia Budianita, Iwan Iskandar, 2020).

Beberapa teknik augmentasi yang umum diterapkan, sebagaimana diuraikan oleh Toyib et al. (2024), meliputi:

1. Rotasi (*Rotation*): Citra diputar berdasarkan sudut tertentu untuk menghasilkan variasi orientasi.
2. Pergeseran Lebar (*Width Shift*): Posisi objek dalam citra digeser secara horizontal.

3. Pergeseran Tinggi (*Height Shift*): Posisi objek dalam citra digeser secara vertikal.
4. Geser (*Shear*): Sebagian piksel citra digeser pada satu sumbu, sehingga menciptakan efek miring.
5. Perbesaran (*Zoom*): Sebagian area citra diperbesar untuk menciptakan variasi skala objek.
6. Pembalikan Horizontal (*Horizontal Flip*): Citra dicerminkan secara horizontal, menciptakan perspektif baru.

2.2.13 Confusion Matrix

Confusion matrix merupakan alat evaluasi fundamental untuk mengukur kinerja sebuah model klasifikasi. Metrik ini disajikan dalam bentuk tabel yang merangkum hasil prediksi model dengan cara membandingkannya terhadap label data yang sebenarnya atau *ground truth* (Normawati & Prayogi, 2021). Matriks ini memiliki empat komponen utama:

- a. True Positive (TP): Jumlah data positif yang berhasil diprediksi dengan benar oleh model.
- b. True Negative (TN): Jumlah data negatif yang berhasil diprediksi dengan benar oleh model.
- c. False Positive (FP): Disebut juga "Kesalahan Tipe I", yaitu jumlah data negatif yang keliru diprediksi sebagai positif.
- d. False Negative (FN): Disebut juga "Kesalahan Tipe II", yaitu jumlah data positif yang keliru diprediksi sebagai negatif.
- e. Berdasarkan nilai-nilai dari *confusion matrix*, beberapa metrik kuantitatif dapat dihitung untuk mengevaluasi performa model dari berbagai aspek:

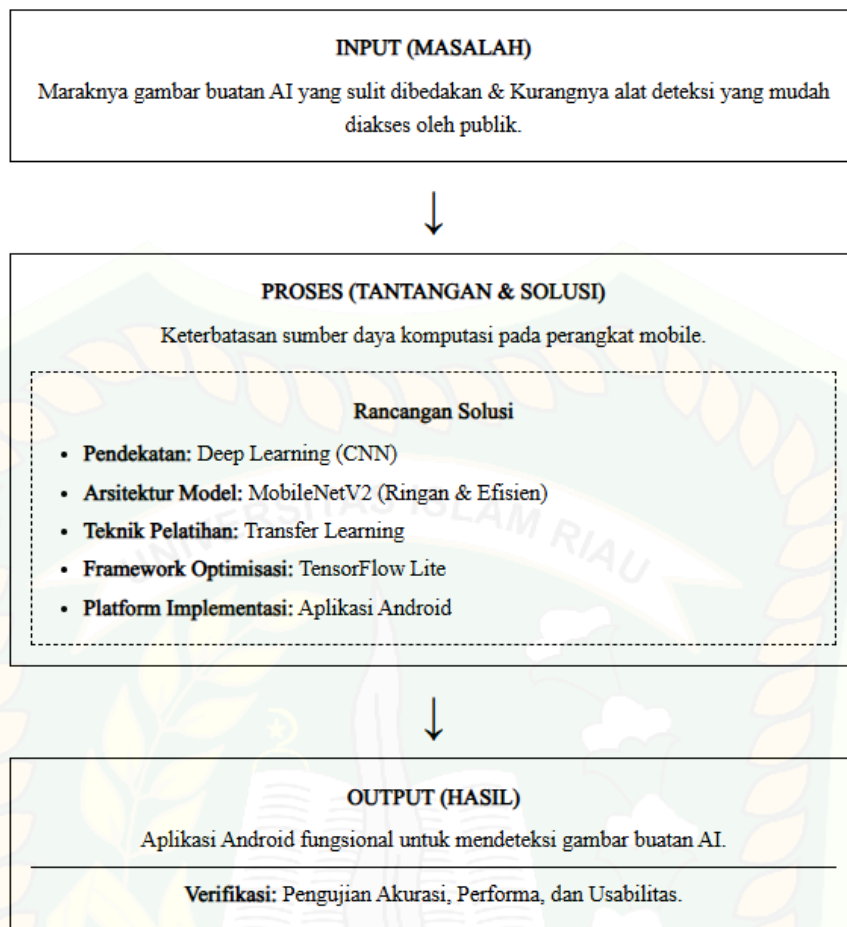
1. Akurasi (*Accuracy*): Mengukur proporsi total prediksi yang benar ($TP + TN$) dari keseluruhan data.
2. Presisi (*Precision*): Mengukur tingkat ketepatan dari prediksi positif yang dibuat oleh model ($TP / (TP + FP)$). Metrik ini menjawab pertanyaan: Dari semua yang diprediksi positif, berapa persen yang benar-benar positif?.
3. Daya Ingat (*Recall* atau Sensitivitas): Mengukur kemampuan model untuk mengidentifikasi semua kasus positif yang ada ($TP / (TP + FN)$). Metrik ini menjawab pertanyaan: "Dari semua kasus yang sebenarnya positif, berapa persen yang berhasil terdeteksi?.
4. Skor F1 (*F1 Score*): Rata-rata harmonik dari *precision* dan *recall*, yang memberikan skor tunggal yang menyeimbangkan kedua metrik tersebut. Sangat berguna pada kasus di mana distribusi kelas tidak seimbang.

2.3 Kerangka Pemikiran

Kerangka pemikiran penelitian ini berawal dari masalah utama yaitu maraknya penyebaran gambar buatan AI yang sangat realistis, yang menimbulkan tantangan bagi masyarakat umum untuk memverifikasi keaslian konten visual. Kebutuhan akan alat deteksi yang cepat, akurat, dan mudah diakses menjadi sangat mendesak. Namun, terdapat kendala teknis yang signifikan: model *deep learning* yang mampu mendeteksi artefak digital yang halus biasanya membutuhkan sumber daya komputasi yang besar, sementara perangkat yang paling mudah diakses oleh masyarakat adalah *smartphone* dengan keterbatasan daya proses dan memori.

Untuk menjawab tantangan tersebut, penelitian ini mengusulkan sebuah solusi yang terstruktur. Pendekatan yang dipilih adalah *Deep Learning* dengan arsitektur *Convolutional Neural Network* (CNN) karena kemampuannya yang terbukti dalam analisis citra. Guna mengatasi kendala perangkat keras, dipilih arsitektur model yang ringan dan efisien yaitu *MobileNetV2*. Untuk mempercepat proses pelatihan dan meningkatkan akurasi tanpa memerlukan *dataset* yang masif, diterapkan teknik *Transfer Learning*.

Setelah model berhasil dilatih untuk dapat membedakan gambar asli dan buatan AI, langkah selanjutnya adalah *optimisasi* agar model dapat berjalan secara optimal di platform Android. Untuk itu, digunakanlah *framework* TensorFlow Lite (*TFLite*) yang akan mengonversi model menjadi format yang lebih ringkas dan cepat. Hasil akhir dari rangkaian proses ini adalah sebuah Aplikasi Android fungsional yang kinerjanya akan diukur melalui pengujian akurasi, kecepatan inferensi, dan usabilitas untuk membuktikan bahwa solusi yang diusulkan efektif dan efisien.



Gambar 2. 3 Kerangka Pemikiran.

BAB III

METODOLOGI PENELITIAN

3.1 Jenis Penelitian

Penelitian ini menggunakan pendekatan eksperimental. Penelitian ini berfokus pada perancangan, pembangunan, dan pengujian eksperimental sebuah sistem deteksi gambar buatan AI untuk mengukur performanya secara objektif.

3.2 Objek Penelitian

Objek dari penelitian ini adalah:

1. Model *Deep Learning*: Model klasifikasi biner yang dikembangkan menggunakan arsitektur *MobileNetV2* dengan teknik *transfer learning* untuk dapat membedakan antara gambar asli (nyata) dan gambar buatan AI.
2. Aplikasi Android: Perangkat lunak yang dikembangkan sebagai platform untuk mengimplementasikan model deteksi, sehingga dapat digunakan oleh pengguna secara praktis pada perangkat *smartphone*.

3.3 Alat dan Bahan Penelitian

3.3.1 Spesifikasi Kebutuhan Hardware

Perangkat keras (*hardware*) yang digunakan dalam pembuatan sistem aplikasi deteksi gambar buatan *AI* adalah :

Tabel 3. 1 Spesifikasi Kebutuhan *Hardware*

No.	Perangkat	Spesifikasi	Fungsi
1	Laptop	Lenovo Ideapad Gaming 3 (Intel Core i5, 16 GB RAM)	Pengembangan kode, pelatihan model, dan kompilasi aplikasi.
2	Cloud GPU	Google Colaboratory	Pelatihan model deep learning yang membutuhkan akselerasi GPU.
3	Smartphone	POCO F6 (Snapdragon 8s Gen 3, 12 GB RAM)	Pengujian fungsionalitas dan performa aplikasi Android.

3.3.2 Spesifikasi Kebutuhan Software dan Data

Perangkat lunak (*software*) yang digunakan dalam pembuatan sistem aplikasi deteksi gambar buatan *AI* adalah :

Tabel 3. 2 Spesifikasi Kebutuhan Software dan Data

NO	Komponen	Nama	Fungsi
1	Platform Pelatihan	Google Colab	Lingkungan interaktif untuk menulis dan mengeksekusi kode Python
2	Framework AI	TensorFlow, TensorFlow Lite, Keras	Untuk membangun, melatih, mengoptimalkan, dan menjalankan model.
3	Model Arsitektur	MobileNetV2 (pretrained on ImageNet)	Arsitektur Model yang digunakan
4	Bahasa Pemrograman	Python	Python untuk skrip pelatihan AI,

5	Dataset	140k Real and Fake Faces, AI, Human Art Classification, dan 196 foto hasil dari kamera ponsel	Sebagai data latih dan validasi untuk model deteksi gambar.
---	---------	--	--



3.4 Alur Penelitian

3.4.1 Pengumpulan Data

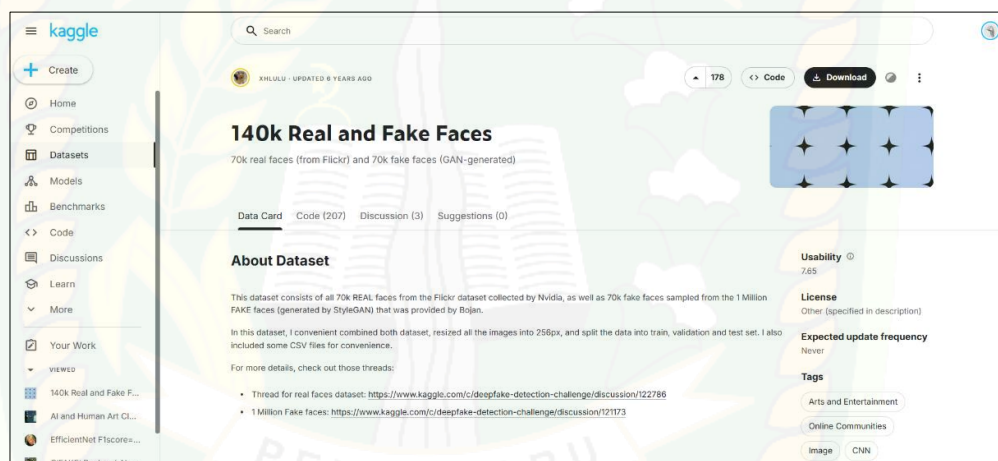
Populasi dalam penelitian ini terdiri dari citra gambar Real dan Fake. Data penelitian diperoleh melalui dua sumber, yaitu data primer dan data sekunder. Data primer dikumpulkan secara langsung oleh peneliti melalui proses pengambilan dan pembuatan citra secara mandiri, baik berupa gambar yang dihasilkan oleh model AI maupun gambar asli dari sumber terbuka.

Sementara itu, data sekunder diperoleh dari situs Kaggle, yang merupakan salah satu platform penyedia dataset terbuka terbesar yang banyak digunakan oleh peneliti dan praktisi untuk mengakses berbagai jenis data, termasuk citra gambar yang dihasilkan oleh AI (AI-generated images) serta gambar non-AI (original images).

Sampel penelitian terdiri atas 2.000 gambar Fake dan 2.000 gambar Real yang diambil dari kombinasi dataset primer dan sekunder tersebut. Pemilihan dataset dilakukan dengan mempertimbangkan aspek variasi jenis gambar, resolusi, dan kategori, sehingga sampel yang digunakan dapat mewakili populasi secara representatif dan mendukung proses pelatihan model deteksi secara optimal. Tahap ini berfokus pada pengumpulan dan persiapan seluruh aset yang dibutuhkan untuk pembangunan model dan aplikasi. Aset utama yang dikumpulkan adalah sebagai berikut:

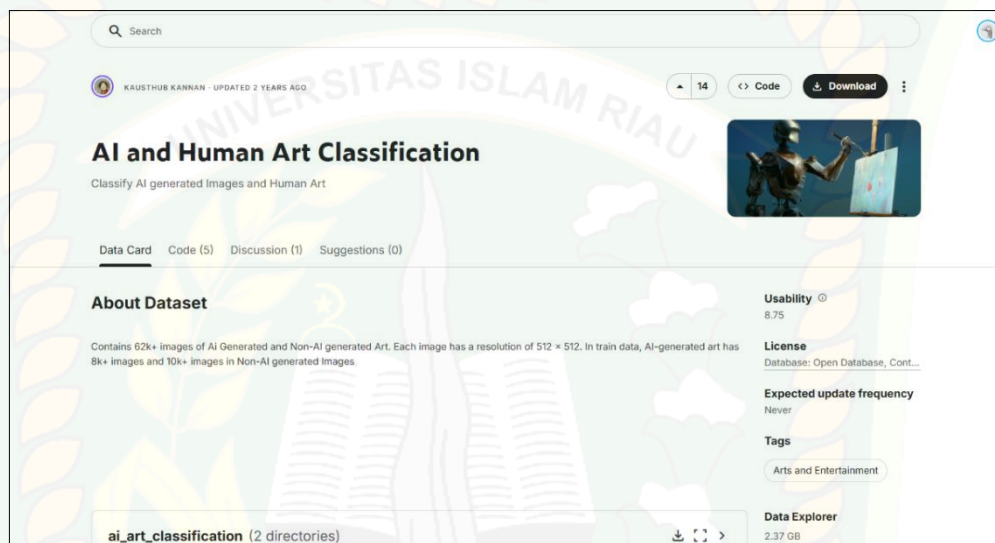
1. Dataset Publik dari Kaggle :

- a. 140k Real and Fake Faces: Sebagai fondasi utama pelatihan, digunakan dataset "140k Real and Fake Faces" yang bersumber dari platform open source Kaggle. Dataset ini berisi 70.000 gambar wajah asli dan 70.000 gambar wajah buatan yang dihasilkan oleh model Generative Adversarial Networks (GAN), khususnya arsitektur StyleGAN. Penggunaan dataset ini bertujuan untuk melatih model secara fundamental dalam mengenali artefak digital dan pola khas yang membedakan citra buatan AI dari citra asli.



Gambar 3. 1 Dataset yang digunakan dari sumber Open Source (Kaggle)

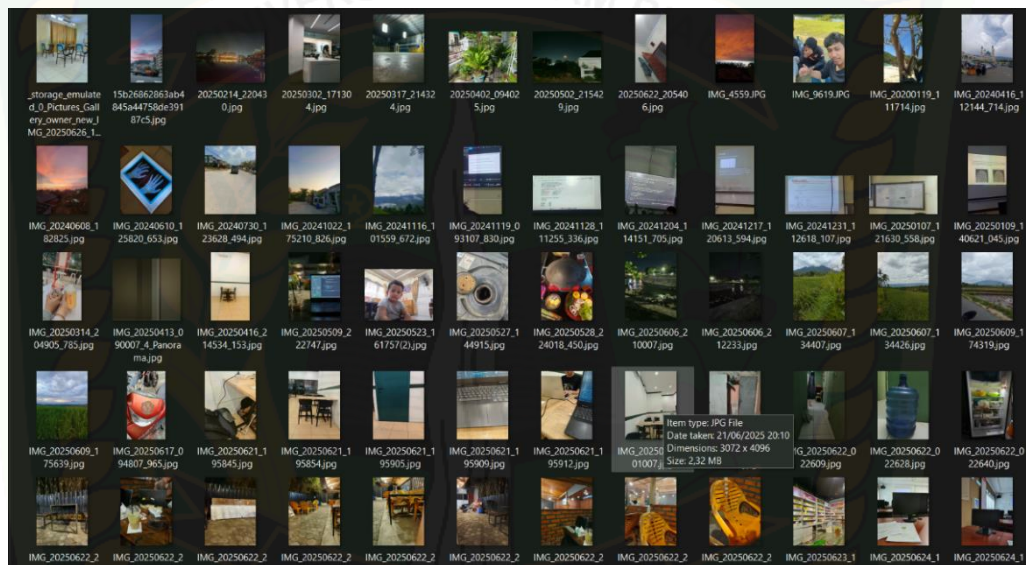
- b. AI and Human Art Classification: Untuk memperluas kemampuan model agar tidak hanya terbatas pada deteksi wajah fotorealistis, penelitian ini juga memanfaatkan dataset "AI and Human Art Classification". Dataset ini menambahkan variasi data berupa karya seni yang dibuat oleh manusia dan yang dihasilkan oleh AI, sehingga melatih model untuk mengenali pola generatif dalam konteks yang lebih luas.



Gambar 3. 2 Dataset yang digunakan dari sumber *Open Source* (Kaggle)

2. Dataset Orisinal Foto Asli dari Kamera Ponsel:

Untuk menambah koleksi gambar asli dan menjamin model berfungsi secara optimal pada foto dari dunia nyata, gambar ini difoto langsung menggunakan kamera ponsel. Foto-foto tersebut diambil dalam berbagai keadaan pencahayaan dan beragam latar belakang. Penambahan data ini penting untuk meningkatkan representasi gambar "REAL" yang otentik dan tidak hanya bergantung pada gambar yang tersedia secara online.



Gambar 3. 3 Dataset Foto Real yang diambil dari kamera Smartphone

Dataset yang digunakan merupakan gabungan dari beberapa sumber untuk memastikan keragaman data . Total sampel penelitian terdiri atas 4.000 gambar *Fake* (buatan AI) dan 4.000 gambar *Real* (asli). Sumber dataset meliputi:

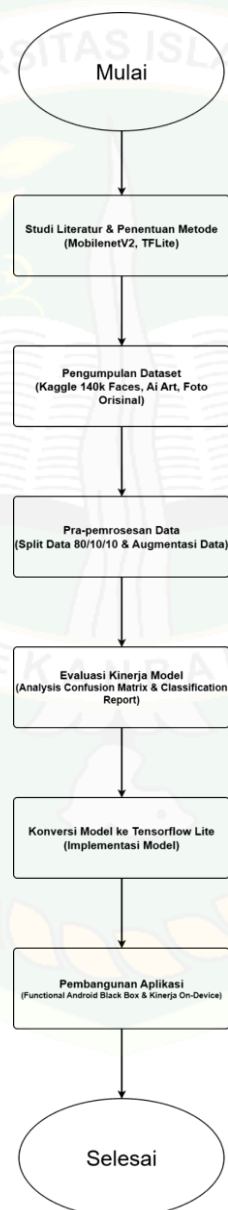
Tabel 3. 3 Jumlah Pembagian Dataset

Sumber Dataset	Kelas Gambar	Jumlah Sampel
Kaggle 140k Real and Fake Faces	FAKE	1000
Kaggle 140k Real and Fake Faces	REAL	804
Kaggle AI and Human Art Classification	FAKE	1000
Kaggle AI and Human Art Classification	REAL	1000
Foto Orisinal (Kamera HP)	REAL	196
TOTAL AKHIR	2000 FAKE & 2000 REAL	4.000

3.4.2 Metode Analisis Data

Untuk memberikan gambaran yang jelas, penelitian ini akan mengikuti alur metodologi yang sistematis. Tahapan ini dimulai dari studi literatur, pengumpulan dataset, pra-pemrosesan data, perancangan dan pelatihan model, konversi model ke TensorFlow Lite, hingga implementasi dan pengujian pada aplikasi Android.

Langkah – Langkah Penelitian :



Gambar 3. 4 Flowchart Penelitian

3.5 Pra–Pemrosesan Data

Analisis data dalam penelitian ini dilakukan menggunakan Google Colab sebagai platform komputasi utama dengan implementasi dalam bahasa pemrograman Python. Proses ini memanfaatkan beberapa pustaka (library) kunci: TensorFlow digunakan untuk membangun arsitektur model *Convolutional Neural Network* (CNN), dengan Keras sebagai *high-level API* untuk menyederhanakan proses pemodelan. Untuk manipulasi data numerik dan visualisasi hasil, digunakan pustaka NumPy dan Matplotlib.

3.5.1 Pra-pemrosesan dan Pemisahan Data (Split Data)

Sebelum memasuki tahap pelatihan, seluruh aset citra digital harus melewati fase standarisasi data. Proses ini meliputi penyesuaian dimensi spasial (*resizing*) citra menjadi format baku 224x224 piksel agar kompatibel dengan *input layer* arsitektur MobileNetV2. Selanjutnya, dilakukan normalisasi intensitas piksel ke dalam rentang skala $[0, 1]$ guna mempercepat konvergensi model. Dataset yang telah terstandarisasi kemudian didistribusikan ke dalam tiga partisi terpisah, yakni data latih, validasi, dan uji, untuk menjamin objektivitas evaluasi.

1. Data Latih (80%): Digunakan untuk melatih model.
2. Data Validasi (10%): Digunakan untuk memantau kinerja model pada setiap *epoch* dan menghindari *overfitting*.
3. Data Uji (10%): Digunakan untuk mengevaluasi kinerja akhir model pada data yang belum pernah dilihat sebelumnya.

```
# Import library yang diperlukan
import splitfolders as sf
import os

# Path awal dataset
base_dir = '/tmp/Image-Classifier-main/Dataset/REAL FAKE'

# Path untuk menyimpan dataset yang telah dibagi
output_dir = '/tmp/Filter Dataset'

# Memastikan direktori output ada atau belum, jika belum, buat direktori
if not os.path.exists(output_dir):
    os.makedirs(output_dir)

# Menggunakan split-folders untuk membagi dataset menjadi train, validation, dan test
sf.ratio(
    input=base_dir,
    output=output_dir,
    seed=None,
    ratio=(0.8, 0.1, 0.1), # 80% train, 10% validation, 10% test
    group_prefix=None # Tidak menggunakan grouping
)
```

Gambar 3. 5 Implementasi Kode untuk Pembagian Dataset (Split Folders) Menjadi Data Latih, Validasi, dan Uji.

3.5.2 Augmentasi Data

Untuk memperkaya variasi data latih, mencegah *overfitting*, dan membuat model lebih tangguh (*robust*), diterapkan teknik augmentasi data . Teknik ini dilakukan hanya pada 80% data latih. Menggunakan *ImageDataGenerator* dari *TensorFlow*, gambar-gambar pelatihan akan dimodifikasi secara acak dengan berbagai transformasi , meliputi:

1. `rotation_range` (Rotasi gambar)
2. `width_shift_range` dan `height_shift_range` (Pergeseran horizontal & vertikal)
3. `shear_range` (Efek miring/geser)
4. `zoom_range` (Perbesaran gambar)
5. `horizontal_flip` (Membalik gambar secara horizontal)

Data validasi dan data uji tidak diaugmentasi, hanya dinormalisasi, agar evaluasi model tetap objektif.

```

from tensorflow.keras.preprocessing.image import ImageDataGenerator

# Augmentasi gambar pada set pelatihan
train_datagen = ImageDataGenerator(
    rescale=1. / 255.,
    width_shift_range=0.2,
    height_shift_range=0.2,
    rotation_range=30,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    brightness_range=(0.8, 1.2), # Mengatur pencahayaan gambar
    fill_mode='nearest' # Mengisi pixel yang hilang setelah transformasi
)

validation_datagen = ImageDataGenerator(rescale=1.0/255.0)

test_datagen = ImageDataGenerator(rescale=1.0/255.0)

# Menggunakan ImageDataGenerator untuk memuat gambar dari direktori dan menerapkan normalisasi dan augmentasi
train_generator = train_datagen.flow_from_directory(
    train_dir,
    target_size=(224, 224),
    batch_size=32,
    class_mode='categorical'
)

validation_generator = validation_datagen.flow_from_directory(
    validation_dir,
    target_size=(224, 224),
    batch_size=32,
    class_mode='categorical'
)

test_generator = test_datagen.flow_from_directory(
    test_dir,
    target_size=(224, 224),
    batch_size=32,
    class_mode='categorical'
)

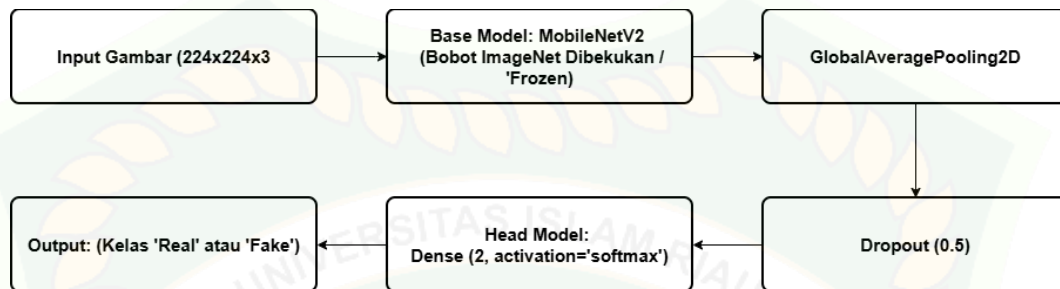
Found 11478 images belonging to 2 classes.
Found 1434 images belonging to 2 classes.
Found 1436 images belonging to 2 classes.

```

Gambar 3. 6 Implementasi Kode Augmentasi Data Menggunakan *ImageDataGenerator*.

3.5 Perancangan Arsitektur dan Pelatihan Model

Model yang dikembangkan menggunakan pendekatan *transfer learning* dengan arsitektur sebagai berikut:



1. **Base Model:** Menggunakan model MobileNetV2 dengan bobot yang telah dilatih pada dataset ImageNet . Lapisan konvolusional dari model ini dibekukan (*frozen*) agar tidak berubah saat pelatihan awal.

```

#membuat model
import tensorflow as tf
from tensorflow.keras.applications import MobileNetV2
from tensorflow.keras import layers
from tensorflow.keras import Model
from tensorflow.keras.optimizers import Adam

#load pretrained model
pretrained_model = MobileNetV2(input_shape=(224, 224, 3),
                                include_top=False,
                                weights='imagenet')

# Freeze sebagian, unfreeze mulai dari blok ke-100 ke atas
for layer in pretrained_model.layers[:100]:
    layer.trainable = False
for layer in pretrained_model.layers[100:]:
    layer.trainable = True
  
```

Gambar 3. 7 Rancangan Arsitektur Model *Transfer Learning* Menggunakan *MobileNetV2*.

2. **Head Model:** Di atas *base model*, ditambahkan lapisan-lapisan kustom, yaitu: `GlobalAveragePooling2D` (mereduksi dimensi), `Dropout(0.5)` (mencegah *overfitting*), dan `Dense(2, activation='softmax')` sebagai lapisan output untuk klasifikasi dua kelas (Real dan Fake) .

```
#menambahkan layer model
x = layers.GlobalAveragePooling2D()(pretrained_model.output)
x = layers.Dense(128, activation='relu')(x)
x = layers.Dropout(0.5)(x)
output = layers.Dense(2, activation='softmax')(x)
```

Gambar 3. 8 Implementasi Kode Penambahan Head Model Kustom (`GlobalAveragePooling`, `Dense`, `Dropout`).

3. **Proses Pelatihan (*Fine-Tuning*):** Pelatihan dilakukan dengan strategi *fine-tuning* . Setelah beberapa *epoch* awal, beberapa lapisan teratas dari *base model* `MobileNetV2` "dicairkan" (*unfrozen*) dan dilatih kembali bersama dengan *head model* menggunakan *learning rate* yang sangat kecil ($1e-5$) . Hal ini bertujuan untuk menyesuaikan fitur-fitur yang telah dipelajari agar lebih relevan dengan tugas deteksi gambar AI.

```
#menggabungkan model
model = Model(pretrained_model.input, output)

# Menampilkan ringkasan model
model.summary()

# Kompilasi ulang sebelum training ulang
model.compile(
    optimizer=Adam(learning_rate=1e-5), # learning rate lebih kecil untuk fine-tuning
    loss='categorical_crossentropy',
    metrics=['accuracy']
)
```

Gambar 3. 9 Implementasi Kode Penggabungan Model dan Kompilasi dengan *Learning Rate* Rendah.

3.7 Konversi Model ke TensorFlow Lite (TFLite)

Setelah model dilatih dan dievaluasi (menggunakan *Classification Report* dan *Confusion Matrix*), model yang disimpan (biasanya dalam format .h5 atau SavedModel) harus dikonversi ke format yang ringan agar dapat berjalan di aplikasi Android. Proses ini menggunakan TensorFlow Lite Converter. Konversi ini akan mengubah arsitektur dan bobot model menjadi file berekstensi .tflite. Tujuannya adalah mengoptimalkan model untuk inferensi *on-device*, mengurangi latensi, serta memperkecil ukuran file aplikasi secara signifikan.

```
import tensorflow as tf

# Path to the model
model_path = "/tmp/model.h5" # path model

# Load the model
model = tf.keras.models.load_model(model_path)

# Konversi model Keras ke TFLite
converter = tf.lite.TFLiteConverter.from_keras_model(model)
tflite_model = converter.convert()

# Simpan model TFLite
with open('model.tflite', 'wb') as f:
    f.write(tflite_model)

print("Model TFLite berhasil dibuat dan disimpan sebagai 'model.tflite'")
```

Gambar 3. 10 Implementasi Kode Konversi Model Keras (.h5) Menjadi *TensorFlow Lite* (.tflite).

BAB IV

HASIL DAN PEMBAHASAN

4.1 Antarmuka Perangkat Lunak

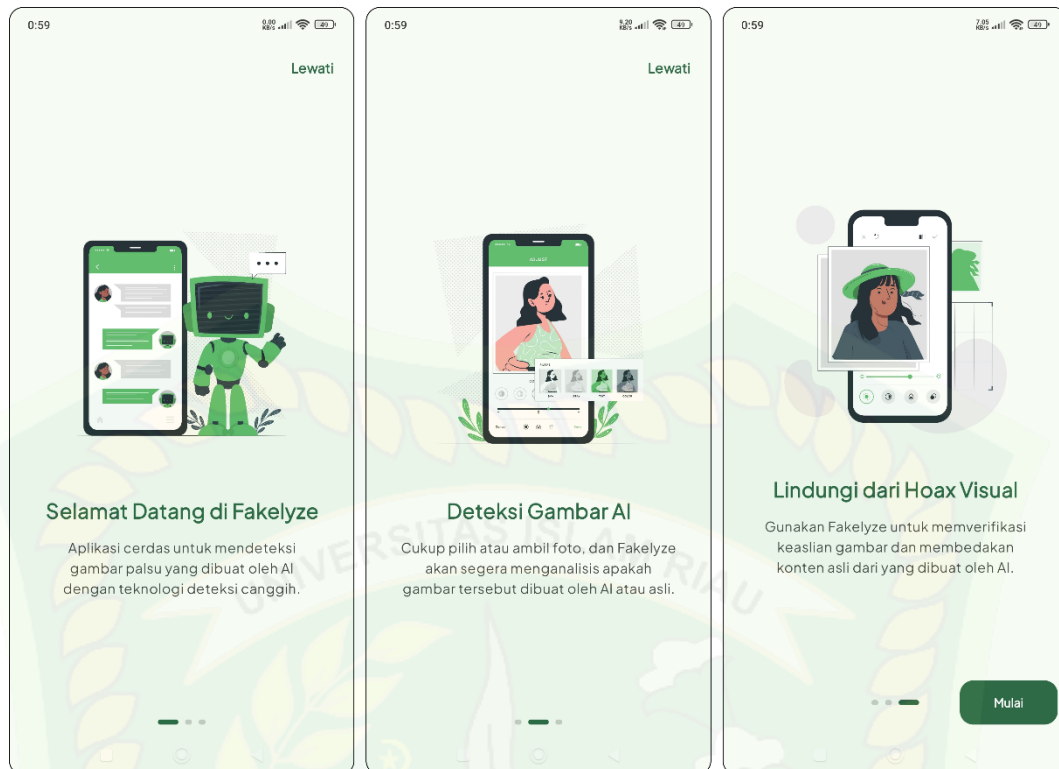
Antarmuka perangkat lunak berfokus pada penciptaan alur pengguna (user flow) yang logis, intuitif, dan efisien. Alur ini memandu pengguna mulai dari pengenalan aplikasi hingga mendapatkan hasil deteksi dan mengelola akun mereka.

4.1.1 Alur Logis Aplikasi

Aplikasi yang dirancang diberi nama Fakelyze, Alur logis aplikasi "Fakelyze" dirancang untuk menangani orientasi pengguna baru, autentikasi, fungsi deteksi inti, dan manajemen akun (termasuk model *freemium*). Alur ini dapat dijabarkan sebagai berikut:

1. Onboarding (Orientasi Pengguna)

Halaman *onboarding* adalah layar pertama yang dilihat pengguna saat membuka aplikasi untuk pertama kalinya. Halaman ini berfungsi untuk memberikan pengenalan singkat mengenai fungsi utama dan manfaat aplikasi, yaitu sebagai alat bantu untuk mendeteksi keaslian gambar. Desainnya dibuat minimalis dengan ilustrasi yang relevan, judul yang jelas, dan deskripsi singkat. Terdapat satu tombol aksi utama (*call-to-action*) yang akan mengarahkan pengguna ke halaman pendaftaran atau masuk.



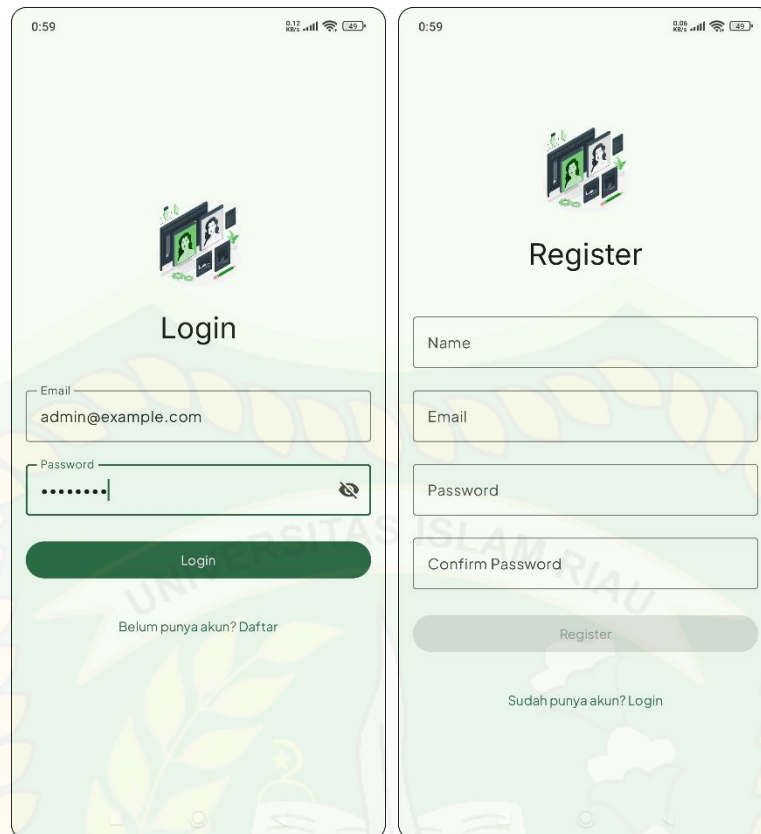
Gambar 4. 1 Tampilan Halaman Onboarding Aplikasi Fakelyze.

2. Autentikasi Pengguna

Halaman ini menyediakan fungsionalitas *autentikasi* bagi pengguna.

1. Halaman *Login*: Pengguna yang sudah memiliki akun dapat masuk dengan memasukkan email dan kata sandi. Terdapat juga opsi "Lupa Kata Sandi" untuk pemulihan akun.
2. Halaman *Register*: Pengguna baru dapat membuat akun dengan mengisi beberapa data dasar seperti nama, email, dan kata sandi.

Desain kedua halaman ini dibuat standar dan mudah dikenali untuk mengurangi kebingungan pengguna.



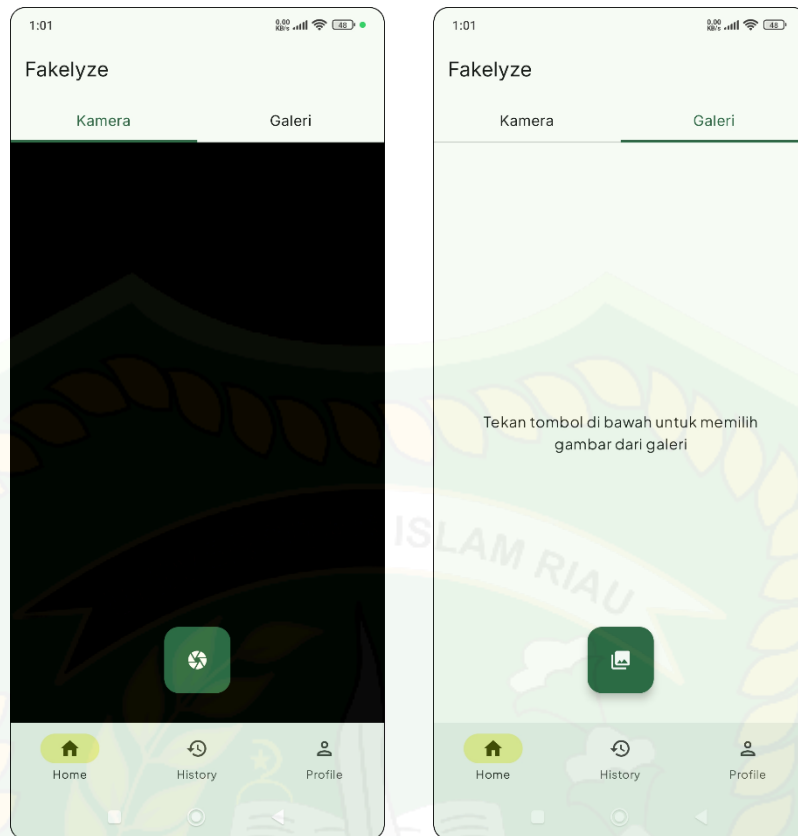
Gambar 4. 2 Tampilan Halaman *Login* dan *Register* untuk *Autentikasi Pengguna*.

3. Halaman Utama (Menu Navigasi Inti)

Setelah berhasil masuk, pengguna akan diarahkan ke Halaman Utama. Halaman ini merupakan pusat dari seluruh aktivitas. Desainnya terpusat pada dua tombol aksi utama yang menjadi fungsi inti aplikasi:

1. Tombol "Gunakan Kamera": Membuka antarmuka kamera perangkat untuk mengambil gambar baru yang akan langsung dianalisis.
2. Tombol "Pilih dari Galeri": Membuka galeri penyimpanan perangkat agar pengguna dapat memilih gambar yang sudah ada untuk dianalisis.

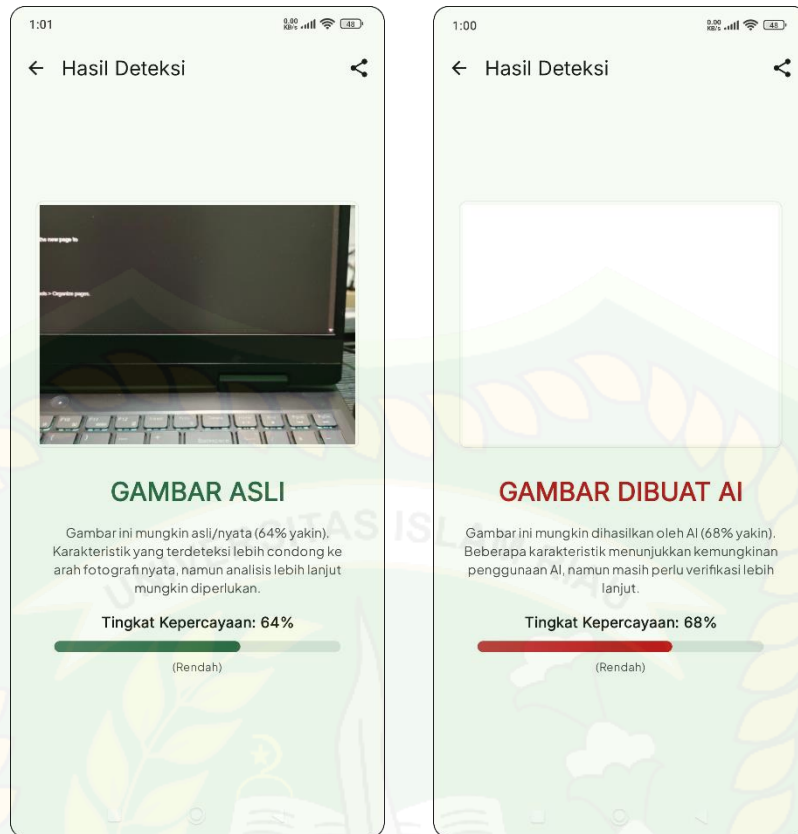
Desain yang sederhana ini bertujuan agar pengguna dapat langsung memahami dan menggunakan fitur utama tanpa hambatan.



Gambar 4. 3 Tampilan Halaman Utama (*Home*) dengan Opsi *Input* Kamera dan Galeri.

4. Alur Proses Deteksi dan Hasil Deteksi

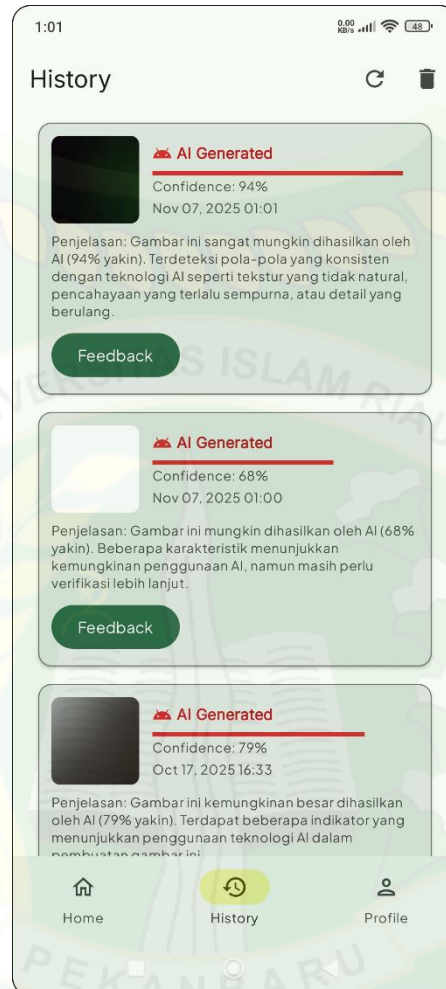
Dari *tab* "Galeri" di Halaman Utama, pengguna dapat menekan tombol ikon galeri untuk memilih gambar dari penyimpanan perangkat. Setelah gambar dipilih, aplikasi akan memproses gambar tersebut menggunakan model MobileNetV2 yang telah diimplementasikan. Aplikasi kemudian menampilkan "Hasil Deteksi".



Gambar 4. 4 Tampilan Halaman Hasil Deteksi untuk Kategori Gambar Asli dan Gambar Buatan AI.

5. Halaman Riwayat (*History*)

Semua hasil pemindaian akan disimpan di Halaman Riwayat.

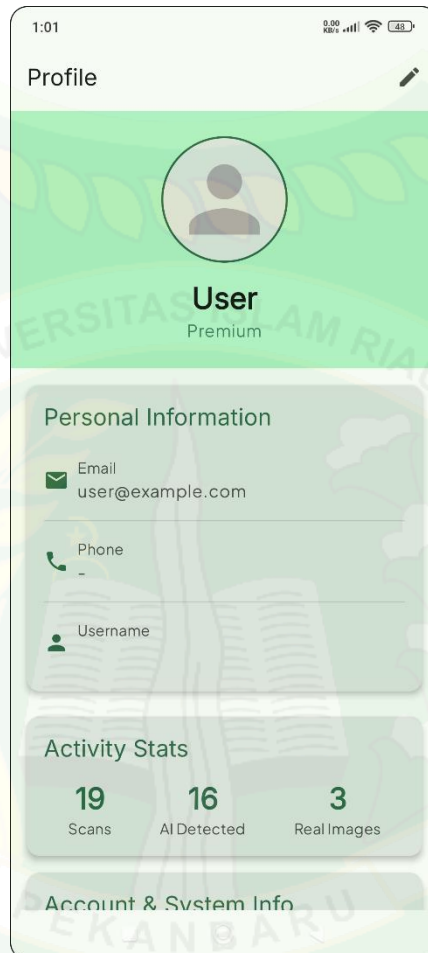


Gambar 4. 5 Tampilan Halaman Riwayat (*History*)
Penyimpanan Hasil Deteksi.

6. Halaman Profile dan Manajemen Akun

Halaman Profil berfungsi sebagai pusat manajemen akun pengguna.

Pengguna dapat melihat "Activity Stats" (jumlah pindaian, deteksi AI, dll.).



Gambar 4. 6 Tampilan Halaman Profil dan Statistik Aktivitas Pengguna.

4.2 Hasil Penelitian

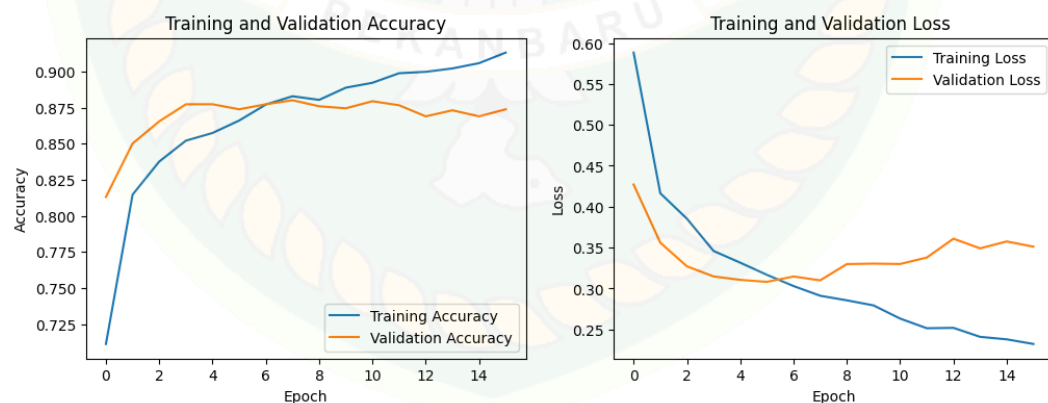
Pada sub-bab ini, disajikan seluruh data kuantitatif yang diperoleh dari hasil pengujian model AI dan pengujian fungsional aplikasi. Data ini menjadi dasar untuk analisis dan pembahasan pada sub-bab berikutnya.

4.2.1 Hasil Evaluasi Kinerja Model AI

Setelah model dilatih menggunakan data latih (80%) dan divalidasi (10%) seperti yang dijelaskan pada BAB III, model diuji menggunakan 10% data uji (*test set*) yang belum pernah dilihat sebelumnya. Data uji ini terdiri dari 800 gambar (400 gambar *FAKE* dan 400 gambar *REAL*). Hasil performa model disajikan dalam *Grafik Matrix Evaluasi Model (Accuracy dan Loss)*, *Classification Report*, dan *Confusion Matrix*.

1. Grafik Matrix Evaluasi Model (Accuracy dan Loss)

visualisasi proses pelatihan dilakukan selama 16 epoch. Grafik di sisi kiri menunjukkan pergerakan Akurasi (*Accuracy*), sedangkan grafik di sisi kanan menunjukkan pergerakan Nilai Kerugian (*Loss*).



Gambar 4. 7 Grafik Pergerakan Akurasi dan Loss Selama 16 Epoch.

Berdasarkan Gambar 4.7 di atas, visualisasi proses pelatihan dilakukan selama 16 epoch. Grafik di sisi kiri menunjukkan pergerakan Akurasi (*Accuracy*), sedangkan grafik di sisi kanan menunjukkan pergerakan Nilai Kerugian (*Loss*).

Garis berwarna biru merepresentasikan data latih (*Training*), sedangkan garis berwarna oranye merepresentasikan data validasi (*Validation*). Dari grafik tersebut, dapat dianalisis beberapa poin penting:

- a. Peningkatan Akurasi Garis akurasi pelatihan (biru) menunjukkan tren kenaikan yang konsisten dari sekitar 71% pada awal pelatihan hingga mencapai angka di atas 91% pada *epoch* akhir. Sementara itu, akurasi validasi (oranye) meningkat tajam pada 3 *epoch* pertama dan kemudian stabil di kisaran angka 87-88%. Stabilitas garis validasi ini menandakan bahwa model mampu menggeneralisasi data baru dengan cukup baik dan tidak hanya menghafal data latih.
- b. Penurunan Loss Grafik *Loss* menunjukkan tren penurunan yang signifikan. *Training Loss* (biru) turun drastis dari 0.60 mendekati 0.24, yang berarti tingkat kesalahan model semakin kecil seiring berjalannya waktu pelatihan. *Validation Loss* (oranye) juga mengalami penurunan hingga titik terendah di sekitar *epoch* ke-7 sebelum akhirnya stabil di kisaran 0.35.
- c. Indikasi Kualitas Model Celah (*gap*) antara garis *training* dan *validation* tidak terlalu lebar, dan garis validasi tidak mengalami penurunan drastis di akhir pelatihan. Hal ini mengindikasikan bahwa model berada dalam kondisi bagus, di mana model telah belajar pola dengan baik tanpa mengalami *overfitting* yang parah (kondisi di mana akurasi latihan tinggi tetapi akurasi validasi anjlok).

2. Classification Report Laporan klasifikasi merangkum performa model dalam mengenali setiap kelas. Model mencapai akurasi keseluruhan sebesar 88% pada data uji.

Classification Report				
	precision	recall	f1-score	support
FAKE	0.86	0.91	0.88	718
REAL	0.90	0.85	0.88	718
accuracy			0.88	1436
macro avg	0.88	0.88	0.88	1436
weighted avg	0.88	0.88	0.88	1436

Gambar 4. 8 Classification Report (Precision, Recall, F1-score).

3. Confusion Matrix memvisualisasikan performa model secara lebih rinci.



Gambar 4. 9 Visualisasi Confusion Matrix Hasil Prediksi Model pada Data Uji

Berdasarkan matriks tersebut, diperoleh hasil sebagai berikut:

1. True Positive (FAKE): Model dengan benar mengidentifikasi 91% gambar FAKE sebagai "FAKE".
2. False Negative (FAKE): Model salah mengidentifikasi 9% gambar FAKE sebagai "REAL".
3. True Negative (REAL): Model dengan benar mengidentifikasi 85% gambar REAL sebagai "REAL".
4. False Positive (REAL): Model salah mengidentifikasi 15% gambar REAL sebagai "FAKE".

4.2.2 Hasil Analisis Nilai Ambang Batas (*Threshold Analysis*)

Selain mengukur akurasi secara umum, penelitian ini juga melakukan analisis terhadap *confidence threshold* (ambang batas kepercayaan) untuk menentukan batas probabilitas yang paling optimal dalam mengklasifikasikan gambar sebagai "Real" atau "Fake".

Secara *default*, model klasifikasi menggunakan *threshold* 0.5 (50%), di mana jika probabilitas prediksi $0.5 - 0.6$ maka akan dikategorikan ke kelas tersebut. Namun, dalam konteks keamanan digital, terkadang diperlukan tingkat kepercayaan yang lebih tinggi untuk meminimalkan kesalahan deteksi (*False Positive*).

Pentingnya analisis ini adalah untuk memastikan bahwa model tidak hanya menebak secara acak, melainkan memiliki tingkat keyakinan (*confidence score*) yang tinggi pada sebagian besar prediksi yang benar.

Analisis terhadap data uji menunjukkan pola distribusi keyakinan model sebagai berikut:

Tabel 4. 1 Distribusi Rata-Rata Skor Keyakinan (*Confidence Score*)

Kelas Prediksi	Rata – rata Confidence Score	Interpretasi
<i>Real</i> (Asli)	0.9	Gambar ini sangat mungkin asli/nyata
<i>Real</i> (Asli)	0.7	Gambar ini kemungkinan besar asli/nyata
<i>Fake</i> (Buatan AI)	0.9	Gambar ini sangat mungkin dihasilkan oleh AI
<i>Fake</i> (Buatan AI)	0.7	Gambar ini kemungkinan besar dihasilkan oleh AI

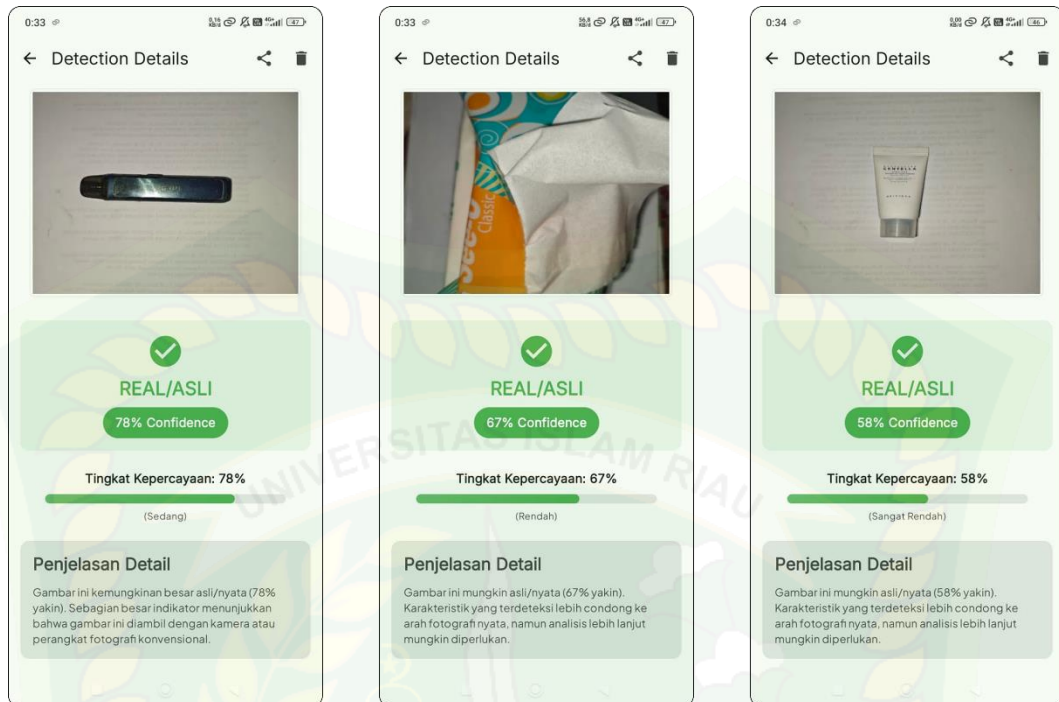
Berdasarkan tabel di atas, rata-rata skor keyakinan model berada jauh di atas ambang batas 0.5. Hal ini mengindikasikan bahwa fitur yang dipelajari oleh MobileNetV2 cukup distingtif (dapat membedakan dengan jelas) antara citra asli dan buatan StyleGAN.

Namun, terdapat sejumlah kecil sampel yang memiliki nilai probabilitas di area ambiguitas (rentang 0.50 - 0.60). Pada kasus-kasus inilah biasanya terjadi kesalahan klasifikasi (*False Positive* atau *False Negative*), yang umumnya disebabkan oleh kualitas gambar yang rendah atau *noise* yang menyerupai artefak GAN.

Dengan tetap mempertahankan *threshold* pada angka 0.5, model berhasil menyeimbangkan nilai *Precision* dan *Recall* (seperti terlihat pada Tabel 4.1 Classification Report), sehingga tidak terlalu agresif dalam memblokir gambar asli, namun tetap efektif menyaring gambar buatan AI.

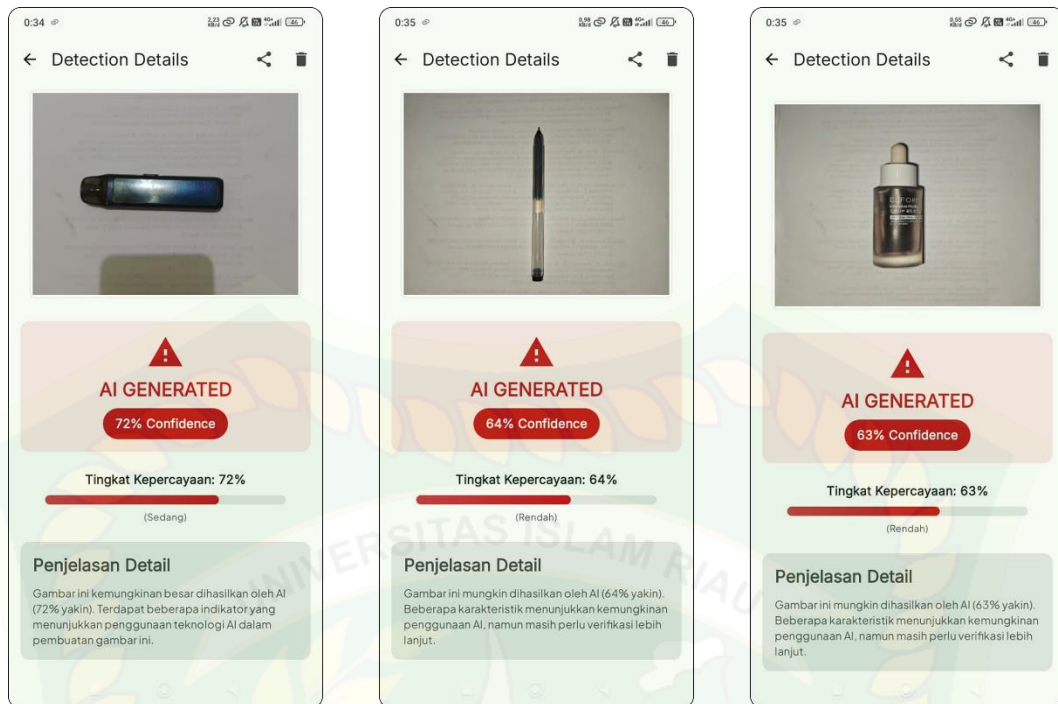
Berdasarkan pengujian langsung menggunakan kamera ponsel terhadap objek nyata, ditemukan bahwa model tidak selalu memberikan keputusan dengan keyakinan penuh (di atas 80% atau 90%). Terdapat sejumlah kasus di mana model memberikan prediksi dengan probabilitas yang berada di dekat ambang batas keputusan 50%.

Hal ini terekam jelas pada hasil pengujian berikut:



Gambar 4. 10 Hasil Deteksi Objek Asli dengan Variasi Skor Keyakinan (78%, 67%, 58%).

Pengujian pada Gambar 4. 10 memperlihatkan kemampuan aplikasi dalam mengenali citra asli (*True Negative*). Meskipun klasifikasi label REAL/ASLI sudah tepat, terdapat gradasi skor keyakinan. Objek dengan fitur distingtif (seperti *pod* hitam) dikenali dengan skor 78%, sedangkan objek dengan warna dominan putih dan minim tekstur (seperti kemasan *lotion*) hanya mencapai skor 58%. Skor 58% ini berada sangat dekat dengan batas ambiguitas, menunjukkan bahwa pencahayaan dan latar belakang polos, dan tingkat ketajaman gambar dapat menurunkan performa deteksi model.



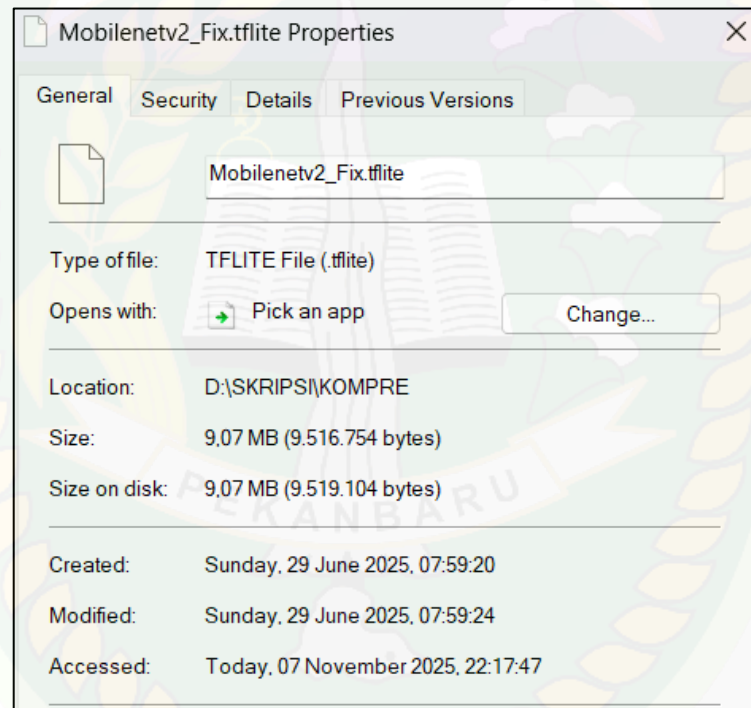
Gambar 4. 11 Tantangan Deteksi: Kesalahan Klasifikasi (*False Positive*) pada Objek Nyata.

Tantangan implementasi *on-device* terlihat pada Gambar 4.11, di mana terjadi kesalahan klasifikasi (*False Positive*). Beberapa objek fisik nyata (pena, botol serum, dan *vape*) terdeteksi sebagai AI GENERATED dengan tingkat keyakinan antara 63% hingga 72%. Kesalahan ini dianalisis terjadi akibat karakteristik permukaan objek yang *glossy* (mengkilap) dan pencahayaan lampu ruangan yang menciptakan efek halus menyerupai karakteristik dari *dataset* citra sintesis GAN (*Generative adversarial Network*). Hal ini menegaskan bahwa faktor lingkungan pengambilan gambar sangat mempengaruhi akurasi inferensi model pada penggunaan nyata.

4.2.3 Hasil Kinerja Implementasi On-Device

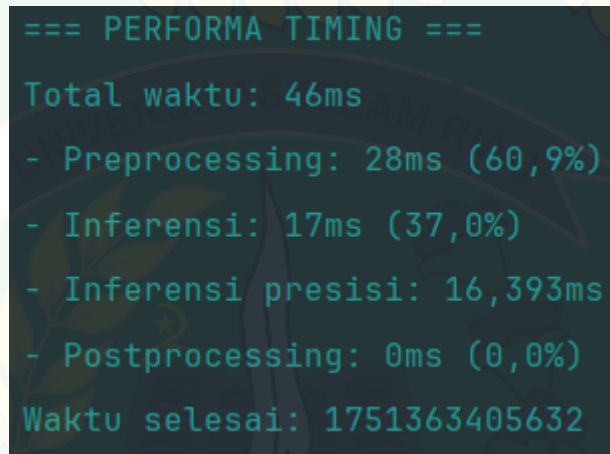
Pengujian ini mengukur efisiensi model setelah dikonversi ke format .tflite dan diimplementasikan pada perangkat Android (POCO F6). Pengujian berfokus pada dua metrik utama: ukuran model dan kecepatan inferensi.

1. Ukuran Model Final Setelah proses pelatihan dan konversi menggunakan *TensorFlow Lite Converter* seperti yang dijelaskan di BAB III, model final berhasil dioptimalkan. Ukuran file Mobilenetv2_Fix.tflite yang diintegrasikan ke dalam aplikasi adalah 9.07 MB.



Gambar 4. 12 Rincian Properti File Model *TensorFlow Lite* (.tflite) Hasil Konversi.

2. Kecepatan Inferensi (Performa Perangkat) Pengujian waktu inferensi *on-device* menunjukkan performa yang sangat cepat. Untuk memproses satu gambar, total waktu yang dibutuhkan adalah 46 milidetik (ms). Waktu ini terbagi menjadi 28ms untuk *Preprocessing* (menyiapkan gambar) dan 17ms untuk *Inferensi* murni (waktu yang dibutuhkan model AI untuk menganalisis).



```
=== PERFORMA TIMING ===  
Total waktu: 46ms  
- Preprocessing: 28ms (60,9%)  
- Inferensi: 17ms (37,0%)  
- Inferensi presisi: 16,393ms  
- Postprocessing: 0ms (0,0%)  
Waktu selesai: 1751363405632
```

Gambar 4. 13 Hasil Log Pengujian Kecepatan Inferensi (*Timing*) pada Perangkat Android.

4.2.4 Hasil Pengujian Fungsional (*Black Box Testing*)

Pengujian fungsional dilakukan untuk memvalidasi bahwa semua fitur perangkat lunak berjalan sesuai dengan skenario pengujian. Seluruh skenario pengujian yang dijalankan memberikan hasil "Berhasil" tanpa ditemukan *bug* fungsional.

Tabel 4. 2 Hasil Pengujian Onboarding & Autentikasi

No.	Skenario Uji	Hasil yang Diharapkan	Hasil Pengujian
1.	Navigasi Onboarding	Pengguna dapat melihat semua layar dan diarahkan ke halaman Login/Register.	Berhasil
2.	Registrasi Akun Baru (Sukses)	Akun berhasil dibuat, aplikasi, mengarahkan pengguna ke halaman Utama.	Berhasil
3.	Login Pengguna (Sukses)	Pengguna berhasil login dan diarahkan ke halaman Utama (Home) aplikasi.	Berhasil
4.	Login Pengguna (Gagal - Data Salah)	Aplikasi menampilkan pesan kesalahan yang jelas dan tetap berada di halaman Login.	Berhasil

Tabel 4. 3 Hasil Pengujian Deteksi Gambar

No.	Skenario Uji	Hasil yang Diharapkan	Hasil Pengujian
1.	Deteksi dari Galeri (Sukses)	Aplikasi berhasil memproses dan menampilkan halaman Hasil Deteksi.	Berhasil
2.	Deteksi dari Kamera (Sukses)	Aplikasi berhasil memproses dan menampilkan halaman Hasil Deteksi.	Berhasil
3.	Penolakan Izin Akses	Aplikasi menampilkan pesan informatif, dan tidak terjadi error.	Berhasil
4.	Input Format File Tidak Valid	Aplikasi menampilkan pesan kesalahan, "Format file tidak didukung".	Berhasil

Tabel 4. 4 Hasil Pengujian Hasil dan Riwayat

No.	Skenario Uji	Hasil yang Diharapkan	Hasil Pengujian
1.	Tampilan Hasil	Halaman hasil menampilkan label, skor, dan penjelasan teks analisis AI secara detail.	Berhasil
2.	Melihat Riwayat (State Kosong)	Halaman menampilkan pesan "Tidak ada riwayat deteksi"	Berhasil

Tabel 4. 5 Hasil Pengujian *Profile*

No.	Skenario Uji	Hasil yang Diharapkan	Hasil Pengujian
1.	Tampilan Profil	Halaman menampilkan status	Berhasil
2.	Deteksi dari Kamera (Sukses)	Aplikasi berhasil memproses dan menampilkan halaman Hasil Deteksi.	Berhasil

4.2.5 Hasil Pengujian User Acceptance Testing (UAT)

Pengujian UAT dilakukan untuk mengukur tingkat kemudahan penggunaan aplikasi dari perspektif pengguna. Metode yang digunakan adalah *User Acceptance Testing* (UAT) Pengujian ini difokuskan pada aspek kemudahan penggunaan, tampilan UI, UX, serta kepuasan pengguna secara keseluruhan dalam menggunakan aplikasi. Yang diujikan kepada (25) responden. Setelah semua kuesioner didapatkan maka selanjutnya dilakukan penghitungan dengan menggunakan skala likert dimana data yang didapatkan dianalisis dengan menghitung rata-rata jawaban berdasarkan skoring.

1. Bobot Jawaban

Pada setiap kategori jawaban memiliki nilai bobot yang berbeda, yang digunakan sebagai dasar dalam proses perhitungan hasil evaluasi pengguna terhadap aplikasi.

Tabel 4. 6 Bobot Jawaban

Kategori Jawaban	Skor
Sangat Setuju (SS)	5
Setuju (S)	4
Netral (N)	3
Tidak Setuju (TS)	2
Sangat Tidak Setuju (STS)	1

2. Statistik Hasil Jawaban Kuesioner

Tabel berikut menyajikan rekapitulasi data respons dari kuesioner yang telah didistribusikan. Distribusi jawaban responden diklasifikasikan berdasarkan lima skala pengukuran yang diterapkan dalam penelitian ini, dengan rentang mulai dari Sangat Setuju hingga Sangat Tidak Setuju.

Tabel 4. 7 Hasil Jawaban Kuesioner

No.	PERTANYAAN	JAWABAN					TOTAL
		SS	S	N	TS	STS	
1.	Tampilan antarmuka aplikasi Pendeteksi Gambar Buatan <i>AI</i> menarik?	5	17	3	0	0	25
2.	Desain tombol dan menu terlihat jelas serta mudah dipahami	8	14	3	0	0	25
3.	Komposisi warna, teks, dan elemen visual dalam aplikasi terlihat cocok dan nyaman dilihat.	8	15	2	0	0	25
4.	Aplikasi menampilkan tata letak yang rapi dan tidak membingungkan.	7	15	3	0	0	25

5.	Perpindahan antar halaman dalam aplikasi berjalan dan responsif.	4	15	6	0	0	25
6.	Setiap tombol lokasi (Seperti Home, History, Profile sesuai)	6	17	2	0	0	25
7.	Waktu loading aplikasi cukup cepat dan tidak mengganggu pengalaman pengguna.	9	11	5	0	0	25
8.	Halaman "Home" memiliki tab "Kamera" dan "Galeri".	8	13	4	0	0	25
9.	Instruksi atau petunjuk interaksi dalam aplikasi mudah dipahami.	8	14	3	0	0	25
10.	Aplikasi terlihat dapat menscan dan membedakan Gambar Asli atau Gambar buatan AI	9	13	3	0	0	25
11.	Seluruh fitur utama dalam aplikasi terlihat berfungsi	7	15	2	1	0	25

	sesuai dengan tujuan.						
12.	Informasi yang ditampilkan melalui visual dan narasi mudah dipahami.	6	19	0	0	0	25
13.	Aplikasi ini dapat menjadi sarana perlindungan digital yang efektif bagi masyarakat agar terhindar dari hoax visual.	7	12	6	0	0	25
14.	Saya merasa puas dengan kinerja dan fitur yang ditawarkan oleh aplikasi Fakelyze.	3	16	6	0	0	25
15.	Aplikasi berhasil membantu saya membedakan antara gambar asli dan gambar buatan AI.	9	14	2	0	0	25
16.	Saya tertarik untuk mengetahui lebih lanjut tentang teknologi deteksi	9	11	5	0	0	25

	gambar AI setelah menggunakan Aplikasi ini.						
17.	Saya bisa merekomendasikan aplikasi ini kepada teman atau keluarga yang ingin terlindungi dari hoax visual.	7	15	3	0	0	25
18.	Waktu yang dibutuhkan aplikasi untuk memproses dan mendeteksi gambar dirasa cukup cepat.	6	14	5	0	0	25
19.	Aplikasi ini perlu dikembangkan lebih lanjut agar semakin menarik dan informatif	10	12	3	0	0	25
20.	Saya tertarik menggunakan aplikasi jika dikembangkan dengan fitur tambahan baru.	10	13	2	0	0	25

3. Jumlah Respon Setiap Kategori

Tabel di bawah ini memperlihatkan distribusi jumlah pilihan responden untuk setiap kategori penilaian, yang mencakup rentang dari Sangat Setuju sampai dengan Sangat Tidak Setuju.

Tabel 4. 8 Jumlah Respon Kategori

Kategori Jawaban	Frekuensi
Sangat Setuju (SS)	146
Setuju (S)	285
Netral (N)	68
Tidak Setuju (TS)	1
Sangat Tidak Setuju (STS)	0

4. Skor Total

Sebelum mengukur tingkat penerimaan pengguna, dilakukan kalkulasi skor total terlebih dahulu. Nilai ini didapat dengan cara menjumlahkan hasil perkalian antara frekuensi kategori jawaban dan bobot skor yang telah ditentukan.

Rumus perhitungan :

$$\text{Total Skor} = (f_1 * 5) + (f_2 * 4) + (f_3 * 3) + (f_4 * 2) + (f_5 * 1)$$

Substitusi Nilai :

$$\begin{aligned}
 \text{Total Skor} &= (146 * 5) + (285 * 4) + (68 * 3) + (1 * 2) + (0 * 1) \\
 &= 730 + 1.140 + 204 + 2 + 0 \\
 &= 2.076
 \end{aligned}$$

Setelah dilakukan substitusi nilai berdasarkan hasil kuesioner, diperoleh total skor sebesar 2.076.

5. Hitung Nilai Rata-rata Skor

Untuk mengetahui tingkat persepsi pengguna terhadap aplikasi, dilakukan perhitungan skor rata-rata. Angka ini merepresentasikan kecenderungan penilaian responden dan akan diklasifikasikan ke dalam kategori jawaban yang tersedia, mulai dari Sangat Setuju hingga kategori lainnya.

Rumus Rata – rata :

$$\begin{aligned} \text{Rata – rata} &= \left(\frac{\text{Total Skor}}{\text{Jumlah Total Respon}} \right) \\ &= \left(\frac{2.076}{500} \right) = 4,15 \end{aligned}$$

Berdasarkan hasil perhitungan tersebut, diperoleh nilai rata-rata nya adalah 4,152

6. Interval Penilaian

Untuk mengkonversi nilai interval Skala Likert ke dalam bentuk persentase, menggunakan rumus.

$$\text{Persentase} = \left(\frac{\text{Nilai Skor}}{\text{Skor Maksimum}} \right) = * 100\%$$

Diketahui :

Skor Maksimum = 5

Skor Minimum = 1

Interval = 0.80

Nilai interval pada dibawah menunjukkan kesimpulan persentase dari setiap soal pada Kuesioner.

Tabel 4. 9 Nilai Interval

Rentang Skor	Kategori Interpretasi	Persentase (%)
1.00 – 1.80	Sangat Tidak Setuju	20% ke bawah
1.81 – 2.60	Tidak Setuju	21% – 40%
2.61 – 3.40	Netral	41% – 60%
3.41 – 4.20	Setuju	61% – 80%
4.21 – 5.00	Sangat Setuju	81% – 100%

Hasil perhitungan menunjukkan nilai rata-rata sebesar **4,15**. Nilai ini berada pada interval **3,41 – 4,20** (61% – 80%), yang menempatkannya dalam kategori **Setuju**. Hal ini mengindikasikan bahwa mayoritas responden memberikan respons yang sangat positif terhadap aspek-aspek yang dinilai, sehingga dapat disimpulkan bahwa aplikasi mendapatkan penerimaan yang sangat baik dari pengguna. Demikian, evaluasi berkelanjutan serta pengembangan berbasis masukan spesifik tetap diperlukan guna mempertahankan dan meningkatkan kualitas aplikasi di masa mendatang.

4.3 Pembahasan

1. Analisis Kinerja Model (AI)

Capaian akurasi pengujian sebesar 88% menjadi indikator kuat bahwa strategi transfer learning yang diterapkan pada MobileNetV2 berjalan efektif. Meskipun merupakan arsitektur yang ringkas, model terbukti mampu menangkap pola-pola artefak mikroskopis yang khas pada citra luaran StyleGAN. Hasil ini menegaskan bahwa deteksi *deepfake* tidak selalu memerlukan model raksasa (*heavyweight models*), melainkan dapat dicapai melalui teknik *fine-tuning* yang presisi pada model yang lebih efisien.

2. Analisis Kinerja Implementasi (On-Device)

Ukuran model final yang hanya 9.07 MB dan waktu inferensi murni 17ms membuktikan bahwa model sangat ringan dan efisien. Ini berhasil menjawab tantangan untuk menjalankan AI canggih di perangkat *mobile* tanpa membebani RAM atau baterai.

3. Keterbatasan Penelitian

Bahwa penelitian ini memiliki keterbatasan, yaitu model hanya dilatih dan diuji pada gambar keluaran StyleGAN . Seperti yang dibatasi pada BAB I, model ini tidak diuji pada model generatif baru seperti Midjourney atau DALL-E 3, sehingga akurasi pada gambar-gambar tersebut tidak diketahui.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil perancangan, implementasi, dan pengujian yang telah dilakukan pada BAB IV, maka ditarik kesimpulan yang menjawab rumusan masalah penelitian sebagai berikut:

1. Tahapan penelitian dimulai dari pengumpulan *dataset* yang terdiri dari gabungan sumber *open source* (Kaggle *140k Real and Fake Faces* dan *AI and Human Art Classification*) serta data primer berupa foto orisinal dari kamera ponsel. Total *dataset* berjumlah 4.000 citra yang terbagi secara seimbang menjadi 2.000 gambar *Real* dan 2.000 gambar *Fake*. Data tersebut kemudian melalui tahap pra-pemrosesan yang meliputi *resizing* ke resolusi 224x224 piksel, normalisasi, serta pembagian data (*splitting*) dengan proporsi 80% data latih, 10% validasi, dan 10% data uji. Khusus pada data latih, diterapkan teknik augmentasi data (seperti rotasi, *zoom*, dan *flip*) untuk meningkatkan variasi dan mencegah *overfitting*. Proses ini dilanjutkan dengan perancangan model menggunakan arsitektur MobileNetV2 berbasis *Transfer Learning* yang dimodifikasi dengan penambahan *head model* kustom (GlobalAveragePooling2D, Dropout 0.5, dan Dense Layer) untuk klasifikasi biner. Model klasifikasi gambar untuk membedakan gambar buatan AI (khususnya *StyleGAN*) dan gambar asli telah berhasil dirancang dan dilatih menggunakan arsitektur MobileNetV2 dengan teknik Transfer

Learning. Model ini dibangun dengan menambahkan *head model* kustom (GlobalAveragePooling2D, Dropout(0.5), dan Dense(2)) di atas *base model* yang telah di-*fine-tuning*.

2. Integrasi model ke dalam sistem Android berjalan optimal melalui konversi TensorFlow Lite, menghasilkan berkas aplikasi yang ringkas (9,07 MB) dan sangat responsif untuk standar penggunaan perangkat seluler.
3. Evaluasi kinerja menunjukkan keseimbangan yang baik antara presisi dan kecepatan, dengan capaian akurasi pengujian sebesar 88% dan kecepatan inferensi 17 milidetik, serta lolos validasi fungsionalitas aplikasi secara menyeluruh.

5.2 Saran

Berdasarkan keterbatasan penelitian yang ditemukan selama proses pengerjaan, berikut adalah beberapa saran untuk pengembangan penelitian selanjutnya:

1. Perluasan Dataset Generatif Model saat ini difokuskan untuk mendeteksi artefak dari *StyleGAN*. Penelitian selanjutnya sangat disarankan untuk memperluas dataset pelatihan dengan menyertakan gambar dari model generatif yang lebih baru dan canggih (seperti Midjourney, DALL-E 3, dan Sora) agar model memiliki kemampuan generalisasi yang lebih luas.
2. Eksplorasi Arsitektur Ringan Lainnya Meskipun MobileNetV2 terbukti sangat efisien, penelitian selanjutnya dapat melakukan studi komparatif dengan arsitektur *mobile-optimized* lainnya, seperti EfficientNet-Lite atau MobileViT, untuk melihat apakah ada arsitektur lain yang dapat memberikan keseimbangan akurasi dan kecepatan inferensi yang lebih baik.
3. Pengembangan Fitur Aplikasi Aplikasi "Fakelyze" dapat dikembangkan lebih lanjut untuk mendukung format file yang lebih beragam (seperti PNG, WEBP) dan fungsionalitas deteksi video atau deteksi URL *real-time* untuk menganalisis gambar langsung dari internet.
4. Pengembangan penelitian selanjutnya disarankan untuk mengintegrasikan teknologi Large Language Model (LLM) berbasis multimodal. Integrasi ini bertujuan untuk melengkapi hasil klasifikasi numerik MobileNetV2 dengan penjelasan deskriptif (natural language explanation). Dengan demikian, aplikasi tidak hanya memberikan label

Asli atau Palsu, tetapi juga mampu memberikan edukasi kepada pengguna mengenai indikasi visual atau artefak spesifik yang ditemukan pada gambar tersebut.

UNIVERSITAS ISLAM RIAU



DAFTAR PUSTAKA

- Android Developers. (2024). *Jetpack Compose UI App Development Toolkit - Android Developers*. <https://developer.android.com/compose>.
- Angeline, A., & Kusniyati, H. (2024). Komparasi Performa VGG19, ResNet50, DenseNet121 dan MobileNetV2 Dalam Mendeteksi Gambar Deepfake. *CESS (Journal of Computer Engineering, System and Science)*, 9(2), 397. <https://doi.org/10.24114/cess.v9i2.58671>
- Aziz, M. A., Rachman, A. S., & Suksmadana, I. M. B. (2024). Pengujian Deteksi Objek Manusia Menggunakan Jetson Nano Dengan Model Ssd Mobilenetv2. *Jurnal Informatika Teknologi Dan Sains (Jinteks)*, 6(4), 830–840. <https://doi.org/10.51401/jinteks.v6i4.4899>
- Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., & Adam, H. (2017). *MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications*. <http://arxiv.org/abs/1704.04861>
- Komputasi, J. I., No, V., Ssd, M., Mobilenet, V., & Model, S. (2020). Pembuatan Aplikasi Deteksi Objek Menggunakan TensorFlow Object Detection API dengan Memanfaatkan SSD MobileNet V2 Sebagai Model Pra - Terlatih. *Jurnal Ilmiah Komputasi*, 19(3), 421–430. <https://doi.org/10.32409/jikstik.19.3.68>
- Muharram, R. F. (2021). Implementasi Artificial Intelligence Untuk Deteksi Masker Secara Realtime Dengan Tensorflow Dan SSD Mobilenet Berbasis Python. *JRKT (Jurnal Rekayasa Komputasi Terapan)*, 1(03), 281–290. <https://doi.org/10.30998/jrkt.v1i03.5832>
- Mutianniza, S. Z., & Suwardoyo, U. (2023). Aplikasi Kamera Cerdas Untuk Deteksi Kendaraan Menggunakan Library Tensorflow. *Jurnal Sintaks Logika*, 3(3), 61–68. <https://doi.org/10.31850/jsilog.v3i3.2589>
- Noprisson, H., Ayumi, V., Purba, M., & Ani, N. (2024). Mobilenet Performance Improvements for Deepfake Image Identification Using Activation Function

- and Regularization. *JITK (Jurnal Ilmu Pengetahuan Dan Teknologi Komputer)*, 10(2), 441–448. <https://doi.org/10.33480/jitk.v10i2.5798>
- Prima, A. (2023). Rancang Bangun Sistem Pendeteksi Aneka Ragam Buah Menggunakan MobileNetv2. *Jurnal Sistim Informasi Dan Teknologi*, 5(2), 208–215. <https://doi.org/10.60083/jsisfotek.v5i2.217>
- Qin, Y., Tang, Q., Xin, J., Yang, C., Zhang, Z., & Yang, X. (2023). A Rapid Identification Technique of Moving Loads Based on MobileNetV2 and Transfer Learning. *Buildings*, 13(2), 1–21. <https://doi.org/10.3390/buildings13020572>
- Rahadi, R. (2024). *Implementasi otomasi black box testing dengan teknik boundary value analysis dan equivalence partitioning studi kasus aplikasi Qarpa PT. Qatros Teknologi April*. <https://eprints.uad.ac.id/64204>
- Tan, M., Chen, B., Pang, R., Vasudevan, V., Sandler, M., Howard, A., & Le, Q. V. (2019). Mnasnet: Platform-aware neural architecture search for mobile. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2019-June, 2815–2823. <https://doi.org/10.1109/CVPR.2019.00293>
- Shadri, M. (2025). *TUGAS AKHIR Deteksi Gambar yang Dihasilkan AI Menggunakan Algoritma Convolutional Neural Network(CNN)*.
- Mustopa, Ali, Agung Sasongko, and Ade Hendini. "DETEKSI RETINOPATI DIABETIK ON-DEVICE MENGGUNAKAN MODEL MOBILENETV2 PADA APLIKASI MOBILE BERBASIS FLUTTER." *Jurnal Informatika dan Teknik Elektro Terapan* 14.1 (2026).
- ROZI, MUHAMMAD FARIS FAHRU. *DETEKSI KANKER KULIT MELANOMA MENGGUNAKAN MOBILENETV2 CONVOLUTIONAL NEURAL NETWORK BERBASIS ANDROID*. Diss. UNIVERSITAS ISLAM SULTAN AGUNG, 2023.
- NURSULISTIO, FERDIAN. "Deteksi Objek Masker Menggunakan Object Detection API dan TensorFlow Lite Model Maker." (2022).

- Mahdy, Irfan Eka. *Klasifikasi Citra Penyakit Pada Daun Tanaman Tomat (Lycopersicum Esculentum Mill.) Menggunakan Metode CNN Berbasis Android*. Diss. Universitas Islam Sultan Agung Semarang, 2025.
- Dong, Ke, et al. "MobileNetV2 model for image classification." *2020 2nd International Conference on Information Technology and Computer Application (ITCA)*. IEEE, 2020.
- Dharmaputra, Azizulhaq, et al. "Aplikasi face mask detection menggunakan neural network mobilenetv2 berbasis android." *Sebatik* 25.2 (2021): 382-389.
- Saputra, Reihan, et al. "Sistem Klasifikasi Alfabet Bahasa Isyarat Indonesia Menggunakan CNN dengan MobileNetV2 berbasis Android." *JUPITER: Jurnal Penelitian Ilmu dan Teknologi Komputer* 17.1 (2025): 237-248.
- Rismayadi, Danny Aidil, and Mohamad Arif Muharam. "PEMANFAATAN MACHINE LEARNING UNTUK OPTIMALISASI LIMBAH DENGAN MODEL MOBILENETV2 PADA APLIKASI ANDROID." *Naratif: Jurnal Nasional Riset, Aplikasi dan Teknik Informatika* 6.1 (2024): 72-78.
- Rahman, Ryan, et al. "Implementasi Mobilenetv2 Untuk Aplikasi Edukasi Sejarah Monumen Nasional Berbasis Computer Vision." *Jurnal Sistem Informasi dan Sistem Komputer* 11.1 (2026): 75-86.
- Widiadnyana, Kadek. *PENGEMBANGAN APLIKASI ANDROID UNTUK KLASIFIKASI JENIS MANGGA LOKAL DI BALI MENGGUNAKAN MOBILENETV2*. Diss. Universitas Pendidikan Ganesha, 2025.
- Putra, Daffa Ananda. *PENGEMBANGAN APLIKASI ANDROID PENGUKUR DERAJAT SANGRAI BIJI KAKAO BERBASIS ANALISIS CITRA MOBILENETV2 DENGAN PEMBELAJARAN TRANSFER LEARNING*. Diss. Universitas Gadjah Mada, 2025.
- Ricky Sebastian, Irawan. *APLIKASI PERANGKAT BERGERAK PENGKLASIFIKASI KESEGERAN DAGING MENGGUNAKAN METODE CONVOLUTIONAL NEURAL NETWORK (CNN) DENGAN ARSITEKTUR MOBILENETV2*. Diss. Sekolah Tinggi Informatika & Komputer Indonesia,

2024.

UNIVERSITAS ISLAM RIAU



SURAT KEPUTUSAN DEKAN FAKULTAS TEKNIK UNIVERSITAS ISLAM RIAU
NOMOR : 0389/KPTS/FT-UIR/SI/2025
TENTANG PENGANGKATAN TIM PEMBIMBING SKRIPSI

DEKAN FAKULTAS TEKNIK

Menimbang :

1. Bahwa untuk menyelesaikan perkuliahan bagi mahasiswa Fakultas Teknik perlu membuat Skripsi.
2. Untuk itu perlu ditunjuk Tim Pembimbing penelitian dan penyusunan Skripsi yang diangkat dengan Surat Keputusan Dekan

Mengingat :

1. Undang-Undang Nomor 12 Tahun 2012 Tentang Pendidikan Tinggi
2. Peraturan Presiden Republik Indonesia Nomor 8 Tahun 2012 Tentang Kerangka Kualifikasi Nasional Indonesia
3. Peraturan Pemerintah Republik Indonesia Nomor 37 Tahun 2009 Tentang Dosen
4. Peraturan Pemerintah Republik Indonesia Nomor 66 Tahun 2010 Tentang Pengelolaan dan Penyelenggaraan Pendidikan
5. Peraturan Menteri Pendidikan dan Kebudayaan Republik Indonesia Nomor 3 Tahun 2020 Tentang Standar Nasional Pendidikan Tinggi
6. Statuta Universitas Islam Riau Tahun 2018
7. Peraturan Universitas Islam Riau Nomor 001 Tahun 2018 Tentang Ketentuan Akademik Bidang Pendidikan Universitas Islam Riau

MEMUTUSKAN

Menetapkan :

1. Mengangkat saudara-saudara yang namanya tersebut dibawah ini sebagai Tim Pembimbing Penelitian & penyusunan Skripsi Mahasiswa/i Fakultas Teknik Program Studi Teknik Informatika

No	Nama	Pangkat	Jabatan
1.	OCTADINO HARYADI, S.Kom., M.Kom.	Asisten Ahli	Pembimbing

2. Mahasiswa yang akan dibimbing :

Nama : Syafwal Gunardi
NPM : 213510057
Program Studi : Teknik Informatika
Jenjang Pendidikan : Strata Satu (S1)
Judul Skripsi : DETEKSI GAMBAR BUATAN AI MENGGUNAKAN TRANSFER LEARNING MOBILENETV2 DAN TENSORFLOW LITE PADA APLIKASI ANDROID

3. Keputusan ini berlaku pada tanggal ditetapkannya dengan ketentuan bila terdapat kekeliruan dikemudian hari segera ditinjau kembali.

Ditetapkan di : Pekanbaru
Pada tanggal : 10 Juni 2025
Dekan,



Dr. DEDDY PURNOMO RETNO, S.T., M.T.
NPK. 100502353



YAYASAN LEMBAGA PENDIDIKAN ISLAM (YLPI) RIAU
UNIVERSITAS ISLAM RIAU

F.A.3.10

Jalan Kaharuddin Nasution No. 113 P. Marpoan Pekanbaru Riau Indonesia – Kode Pos: 28284
Telp. +62 761 674674 Fax. +62 761 674834 Website: www.uir.ac.id Email: info@uir.ac.id

KARTU BIMBINGAN TUGAS AKHIR
SEMESTER GENAP TA 2024/2025

NPM : 213510057
Nama Mahasiswa : SYAFWAL GUNARDI
Dosen Pembimbing : 1. OCTADINO HARYADI S.Kom., M.I.P., M.Kom 2.
Program Studi : TEKNIK INFORMATIKA
Judul Tugas Akhir : DETEKSI GAMBAR BUATAN AI MENGGUNAKAN TRANSFER LEARNING
MOBILENETV2 DAN TENSORFLOW LITE PADA APLIKASI ANDROID
Judul Tugas Akhir (Bahasa Inggris) : AI-GENERATED IMAGE DETECTION USING TRANSFER LEARNING
MOBILENETV2 AND TENSORFLOW LITE ON ANDROID APPLICATION
Lembar Ke :

NO	Hari/Tanggal Bimbingan	Materi Bimbingan	Hasil / Saran Bimbingan	Paraf Dosen Pembimbing
1	6/6/2025	Revisi Bab 1 - 2	Revisi	
2	16/6/2025	Bab 2 – 3	Revisi	
3	24/06/2025	Bab 2 - 3	Revisi	
4	01/07/2025	ACC SEMPRO	ACC	
5	25/10/2025	Bab 4	Revisi	
6	11/11/2025	Bab 4 dan Bab 5, System	Revisi	
7	8/12/2025	ACC KOMPRES	ACC	

Pekanbaru, 08 Desember 2025
Wakil Dekan I/ Ketua Departemen/ Ketua Prodi



(Dr. Ir Des Suryani, M.Sc.)

Catatan :

1. Lama bimbingan Tugas Akhir/ Skripsi maksimal 2 semester sejak TMT SK Pembimbing diterbitkan
2. Kartu ini harus dibawa setiap kali berkonsultasi dengan pembimbing dan HARUS dicetak kembali setiap memasuki semester baru melalui SIKAD
3. Saran dan koreksi dari pembimbing harus ditulis dan diparaf oleh pembimbing
4. Setelah skripsi disetujui (ACC) oleh pembimbing, kartu ini harus ditandatangani oleh Wakil Dekan I/ Kepala departemen/ Ketua prodi
5. Kartu kendali bimbingan asli yang telah ditandatangani diserahkan kepada Ketua Program Studi dan kopiannya dilampirkan pada skripsi.
6. Jika jumlah pertemuan pada kartu bimbingan tidak cukup dalam satu halaman, kartu bimbingan ini dapat di download kembali melalui SIKAD

SURAT KEPUTUSAN DEKAN FAKULTAS TEKNIK UNIVERSITAS ISLAM RIAU
NOMOR : 0021/KPTS/FT-UIR/2026
TENTANG PENETAPAN DOSEN PENGUJI SKRIPSI MAHASISWA FAK. TEKNIK UNIV. ISLAM RIAU

DEKAN FAKULTAS TEKNIK

- Menimbang** : 1. Bahwa untuk menyelesaikan studi S.1 bagi mahasiswa Fakultas Teknik Univ. Islam Riau dilaksanakan Ujian Skripsi/Komprehensif sebagai tugas akhir. Untuk itu perlu ditetapkan mahasiswa yang telah memenuhi syarat untuk ujian dimaksud serta dosen penguji.
2. Bahwa penetapan mahasiswa yang memenuhi syarat dan dosen penguji yang bersangkutan perlu ditetapkan dengan Surat Keputusan Dekan.
- Mengingat** : 1. Undang - Undang Nomor 12 Tahun 2012 Tentang Pendidikan Tinggi
2. Peraturan Presiden Republik Indonesia Nomor 8 Tahun 2012 Tentang Kerangka Kualifikasi Nasional Indonesia
3. Peraturan Pemerintah Republik Indonesia Nomor 37 Tahun 2009 Tentang Dosen
4. Peraturan Pemerintah Republik Indonesia Nomor 66 Tahun 2010 Tentang Pengelolaan dan Penyelenggaraan Pendidikan
5. Peraturan Menteri Pendidikan Nasional Nomor 63 Tahun 2009 Tentang Sistem Penjaminan Mutu Pendidikan
6. Peraturan Menteri Pendidikan dan Kebudayaan Republik Indonesia Nomor 49 Tahun 2014 Tentang Standar Nasional Pendidikan Tinggi
7. Statuta Universitas Islam Riau Tahun 2018
8. Peraturan Universitas Islam Riau Nomor 001 Tahun 2018 Tentang Ketentuan Akademik Bidang Pendidikan Universitas Islam Riau

MEMUTUSKAN

- Menetapkan** : 1. Mahasiswa Fakultas Teknik Universitas Islam Riau yang tersebut namanya dibawah ini :
- Nama : Syafwal Gunardi
- NPM : 213510057
- Program Studi : Teknik Informatika
- Jenjang Pendidikan : Strata Satu (S1)
- Judul Skripsi : Deteksi Gambar Buatan Ai Menggunakan Transfer Learning Mobilenetv2 Dan TensorflowLite Pada Aplikasi Android
2. Penguji Skripsi/Komprehensif mahasiswa tersebut terdiri dari :
1. Octadino Haryadi, S.Kom., M.Kom Sebagai Ketua Merangkap Penguji
2. Anggi Hanafiah, S.Kom., M.Kom Sebagai Anggota Merangkap Penguji
3. Ana Yulianti, S.T., M.Kom. Sebagai Anggota Merangkap Penguji
3. Laporan hasil ujian serta berita acara telah sampai kepada Pimpinan Fakultas selambat-lambatnya 1(satu) bulan setelah ujian dilaksanakan.
4. Keputusan ini mulai berlaku pada tanggal ditetapkannya dengan ketentuan bila terdapat kekeliruan dikemudian hari segera ditinjau kembali.
- KUTIPAN : Disampaikan kepada yang bersangkutan untuk dapat dilaksanakan dengan sebaik-baiknya.

Ditetapkan di : Pekanbaru
Pada Tanggal : 19 Rajab 1447 H
08 Januari 2026 M

Dekan,



Dr. Ir. KURNIA HASTUTI, S.T, M.T

NPK : 1008057102

Tembusan disampaikan :

1. Yth. Rektor UIR di Pekanbaru.
2. Yth. Ketua Program Studi Teknik Informatika FT-UIR
3. Yth. Pembimbing dan Penguji Skripsi
3. Mahasiswa yang bersangkutan
5. Arsip

**Surat ini ditandatangani secara elektronik*



YAYASAN LEMBAGA PENDIDIKAN ISLAM (YLPI) RIAU
UNIVERSITAS ISLAM RIAU
FAKULTAS TEKNIK
PROGRAM STUDI TEKNIK INFORMATIKA

Jalan Kaharuddin Nasution No. 113 P. Marpoyan Pekanbaru Riau Indonesia – Kode Pos: 28284
Telp. +62 761 674674 Website: www.eng.uir.ac.id Email: fakultas_teknik@uir.ac.id

BERITA ACARA UJIAN SKRIPSI

Berdasarkan Surat Keputusan Dekan Fakultas Teknik Universitas Islam Riau, Pekanbaru, tanggal 07 Januari 2026, Nomor: 0021/KPTS/FT-UIR/2026, maka pada hari Kamis tanggal 08 Januari 2026, telah dilaksanakan Ujian Skripsi Program Studi Teknik Informatika Fakultas Teknik Universitas Islam Riau, Jenjang Studi S1, Tahun Akademik 2025/2026 berikut ini.

1. Nama : Syafwal Gunardi
2. NPM : 213510057
3. Judul Skripsi : Deteksi Gambar Buatan Ai Menggunakan Transfer Learning Mobilenetv2 Dan TensorflowLite Pada Aplikasi Android
4. Waktu Ujian : 11.00 WIB s.d. Selesai
5. Tempat Pelaksanaan Ujian : Ruang Sidang Fakultas Teknik UIR

Dengan keputusan Hasil Ujian Skripsi:

~~Lulus~~*/ Lulus dengan Perbaikan*/ ~~Tidak Lulus~~*

* Coret yang tidak perlu.

Nilai Ujian:

Nilai Ujian Angka = 77.5 Nilai Huruf = A-

Tim Penguji Skripsi.

No	Nama	Jabatan	Tanda Tangan
1	Octadino Haryadi, S.Kom., M.Kom	Ketua	1.
2	Anggi Hanafiah, S.Kom., M.Kom	Anggota	2.
3	Ana Yulianti, S.T., M.Kom.	Anggota	3.

Panitia Ujian
Ketua

Octadino Haryadi, S.Kom., M.Kom.
NIDN. 1031109201

Pekanbaru, 08 Januari 2026
Mengetahui,
Dekan Fakultas Teknik

Dr. T. Kurnia Hastuti, S.T., M.T.
NIDN. 1008057102



UNIVERSITAS ISLAM RIAU

FAKULTAS TEKNIK

الجامعة الإسلامية الريوية

Alamat: Jalan Kaharuddin Nasution No.113, Marpoyan, Pekanbaru, Riau, Indonesia - 28284
Telp. +62 761 674674 Email: fakultas_teknik@uir.ac.id Website: www.eng.uir.ac.id

SURAT KETERANGAN BEBAS PLAGIAT

Nomor: 631/A-UIR/5-T/2025

Fakultas Teknik Universitas Islam Riau menerangkan bahwa Mahasiswa/i dengan identitas berikut:

Nama : Syafwal Gunardi
NPM : 213510057
Program Studi : Teknik Informatika
Jenjang Pendidikan : Strata Satu (S1)
Judul Skripsi TA : DETEKSI GAMBAR BUATAN AI MENGGUNAKAN
TRANSFER LEARNING MOBILENETV2 DAN
TENSORFLOW LITE PADA APLIKASI ANDROID

Dinyatakan **Bebas Plagiat**, berdasarkan hasil pengecekan pada Turnitin menunjukkan angka **Similarity Index < 30%** sesuai dengan peraturan Universitas Islam Riau yang berlaku.

Demikian surat keterangan ini dibuat untuk dapat dipergunakan sebagaimana mestinya.

Mengetahui,

Kaprodi. Teknik Informatika

Assoc. Prof. Dr. Ir. Hj. Des Suryani, M.Sc.

Pekanbaru, 17 December 2025 M

26 Jumādil Akhiroh 1447 H

Staff Pemeriksa

Syatra Safira Anjani, S.Pd