# DEEPFAKE FACE DETECTION MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORK (CNN) DENGAN ARSITEKTUR EFFICIENTNET-B0

(Studi Kasus : Klasifikasi Wajah Asli dan Wajah Deepfake (AI-generated) pada Dataset Wajah Orang Indonesia)

**TUGAS AKHIR**

Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh Gelar Sarjana Program Studi Statistika


Disusun Oleh:

Widya Saputri Agustin

22611001



**PROGRAM STUDI STATISTIKA**

**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM**

**UNIVERSITAS ISLAM INDONESIA**

**YOGYAKARTA**



**2026**

# HALAMAN PERSETUJUAN PEMBIMBING

# TUGAS AKHIR

| | | |
|---|---|---|
| Judul | : | Deepfake Face Detection Menggunakan Convolutional Neural Network (CNN) dengan Arsitektur EfficientNet-B0 (Studi Kasus: Klasifikasi Wajah Asli dan Wajah Deepfake (AI-generated) pada Dataset Wajah Orang Indonesia) |
| Nama Mahasiswa | : | Widya Saputri Agustin |
| NIM | : | 22611001 |

**TUGAS AKHIR INI TELAH DIPERIKSA DAN DISETUJUI UNTUK DIUJIKAN**

| | |
|---|---|
| Mengetahui | Yogyakarta, 6 Mei 2026 |
| Ketua Prodi Statistika | Dosen Pembimbing |
| **(Dr. Atina Ahdika, S.Si., M.Si.)** | **(Ayundyah Kesumawati, S.Si., M.Si.)** |

# HALAMAN PENGESAHAN

# TUGAS AKHIR

**DEEPFAKE FACE DETECTION MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORK (CNN) DENGAN ARSITEKTUR EFFICIENTNET-B0**

(Studi Kasus : Klasifikasi Wajah Asli dan Wajah Deepfake (AI-generated) pada Dataset Wajah Orang Indonesia)

**Nama Mahasiswa : Widya Saputri Agustin**

**NIM : 22611001**

**TUGAS AKHIR INI TELAH DIUJIKAN**

**PADA TANGGAL : 1 JULI 2026**

| Nama Penguji | Tanda Tangan |
|---|---|
| 1. Dr. Raden Bagus Fajriya Hakim, S.Si., M.Si. | ..... |
| 2. Dr. techn. Rohmatul Fajriyah, S.Si., M.Si. | ..... |
| 3. Ayundyah Kesumawati, S.Si., M.Si. | ..... |

Mengetahui,

Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam

Universitas Islam Indonesia

**(apt. Saepudin, S.Si., M.Si., Ph.D.)**

# KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَٰنِ الرَّحِيمِ

*Assalamu'alaikum Wr.Wb*

Puji syukur kehadirat Allah SWT atas limpahan rahmat dan karunia-Nya, sehingga penulis dapat menyusun Tugas Akhir dengan judul "*Deepfake Face Detection* Menggunakan *Convolutional Neural Network* (CNN) dengan Arsitektur EfficientNet-B0". Shalawat serta salam tak lupa selalu tercurah kepada Nabi Muhammad SAW beserta keluarga, sahabat, dan para pengikutnya hingga akhir zaman.

Tugas Akhir ini disusun sebagai salah satu syarat untuk memperoleh gelar sarjana Program Studi Statistika, Universitas Islam Indonesia. Penelitian ini bertujuan untuk menganalisis kemampuan model *Convolutional Neural Network* dengan arsitektur EfficientNet-B0 dalam mengklasifikasikan wajah asli dan wajah *deepfake* yang berasal dari citra *AI-generated* pada *dataset* wajah orang Indonesia. Hasil penelitian ini diharapkan dapat memberikan kontribusi dalam pengembangan metode deteksi *deepfake* berbasis citra wajah serta menjadi referensi bagi penelitian selanjutnya dalam bidang keamanan digital dan pengolahan citra.

Dalam proses penyusunan Tugas Akhir ini, penulis mendapatkan banyak bimbingan, dukungan, dan bantuan dari berbagai pihak. Oleh karena itu, penulis ingin menyampaikan terima kasih kepada:

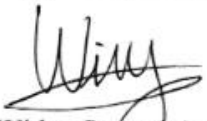
1. Kedua orang tua dan keluarga yang selalu memberikan dukungan moral, spiritual, serta do'a yang tak ada henti-hentinya.
2. Bapak Prof. Riyanto, S.Pd., M.Si., Ph.D., selaku Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Islam Indonesia.
3. Bapak Edy Widodo, Dr., S.Si., M.Si., selaku Ketua Jurusan Statistika beserta seluruh jajarannya.
4. Bapak/Ibu Dosen di Program Studi Statistika, Universitas Islam Indonesia, yang telah memberikan ilmu dan wawasan yang sangat berharga dalam bidang statistika.

5. Ibu Ayundyah Kesumawati, S.Si., M.Si., selaku dosen pembimbing yang telah memberikan waktu, arahan, bimbingan, dan dukungan dalam penyusunan Tugas Akhir ini.
6. Teman-teman dan rekan seperjuangan yang selalu memberikan motivasi dan semangat dalam menyelesaikan Tugas Akhir ini.
7. Semua pihak yang telah membantu, baik secara langsung maupun tidak langsung, dalam proses penyusunan Tugas Akhir ini.

Penulis menyadari bahwa Tugas Akhir ini masih jauh dari kata sempurna, baik dari segi isi maupun penyajian. Oleh karena itu, kritik dan saran yang membangun sangat diharapkan untuk perbaikan di masa mendatang. Semoga Tugas Akhir ini dapat bermanfaat, baik bagi penulis maupun bagi pengembangan penelitian di bidang klasifikasi citra wajah. Akhir kata, semoga Allah SWT senantiasa melimpahkan rahmat dan hidayah-Nya kepada kita semua. Aamiin

*Wassalamu'alaikum Wr.Wb*

Yogyakarta, 4 Mei 2026



Widya Saputri Agustin

# DAFTAR ISI

# DAFTAR TABEL

# DAFTAR GAMBAR

# DAFTAR LAMPIRAN

# PERNYATAAN

Dengan ini saya menyatakan bahwa dalam Tugas Akhir ini tidak terdapat karya karya yang sebelumnya pernah diajukan untuk memperoleh gelar kesarjanaan di suatu Perguruan Tinggi dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan orang lain, kecuali yang diacu dalam naskah ini dan disebutkan dalam daftar pustaka.

Yogyakarta, 4 Mei 2026


Widya Saputri Agustin

# INTISARI

## DEEPFAKE FACE DETECTION MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORK (CNN) DENGAN ARSITEKTUR EFFICIENTNET-B0

(Studi Kasus : Klasifikasi Wajah Asli dan Wajah Deepfake (AI-generated) pada Dataset Wajah Orang Indonesia)

Widya Saputri Agustin
Program Studi Statistika, Fakultas Matematika dan Ilmu Pengetahuan Alam
Universitas Islam Indonesia

Perkembangan pesat *Artificial Intelligence* (AI) membuat gambar hasil generasi AI semakin menyerupai foto autentik dan sulit dibedakan secara visual. Wajah seseorang kini dapat dimanipulasi karena adanya berbagai algoritma berbasis teknologi *deep learning*. Salah satu fenomena yang muncul sebagai konsekuensi dari perkembangan tersebut adalah *deepfake*. Menurut laporan Deepfake Statistics 2025 (AI Fraud Data & Trends), jumlah *file deepfake* secara global meningkat dari sekitar 5,5 juta pada tahun 2023 dan diproyeksikan mencapai lebih dari 8 juta *file* pada tahun 2025. Menurut Kementerian Komunikasi dan Digital (Komdigi) Republik Indonesia, fenomena konten *deepfake* telah meningkat tajam. Data dari *platform* deteksi ancaman AI menunjukkan bahwa jumlah konten *deepfake* meningkat sebesar 550% sejak 2019 dan 90% di antaranya digunakan untuk tujuan berbahaya. Penelitian ini menggunakan *Convolutional Neural Network* (CNN) dengan arsitektur EfficientNet-B0 untuk mengklasifikasikan wajah asli dan *deepfake* pada *dataset* wajah orang Indonesia. Hasil pengujian menunjukkan bahwa model menunjukkan kinerja yang baik dan konsisten dalam mendeteksi wajah *deepfake*. Evaluasi pada data uji menghasilkan akurasi sebesar 98%, dengan rata-rata *F1-score* di atas 0.96 serta nilai AUC pada rentang 0.999 - 1.000. Visualisasi t-SNE menunjukkan bahwa model mampu memisahkan representasi fitur wajah asli dan *deepfake* ke dalam klaster yang terpisah secara jelas. Berdasarkan Grad-CAM, dapat diidentifikasi bahwa keputusan model dipengaruhi oleh analisis tekstur kulit wajah di area pipi dan rahang, dengan kontribusi yang relatif rendah dari elemen latar belakang. Hasil penelitian ini diharapkan dapat menjadi dasar bagi pengembangan sistem deteksi *deepfake* otomatis serta mendukung penguatan keamanan siber berbasis AI dalam mengurangi penyalahgunaan konten manipulatif di Indonesia.

**Kata Kunci** : AI generatif, *deepfake, deep learning*, klasifikasi gambar, CNN

# ABSTRACT


**DEEPFAKE FACE DETECTION USING CONVOLUTIONAL NEURAL NETWORK (CNN) WITH EFFICIENTNET-B0 ARCHITECTURE**
(Case Study : Classification of Real Faces and AI-Generated Deepfake Faces in an Indonesian Face Dataset)

Widya Saputri Agustin
Department of Statistics, Faculty of Mathematics and Natural Sciences
Universitas Islam Indonesia

*The rapid advancement of Artificial Intelligence (AI) has enabled AI-generated images to increasingly resemble authentic photographs, making them difficult to distinguish visually. A person's face can now be manipulated through various algorithms based on deep learning technology. One phenomenon that has emerged as a consequence of this development is deepfake. According to the Deepfake Statistics 2025 (AI Fraud Data & Trends) report, the global number of deepfake files increased from approximately 5.5 million in 2023 and is projected to exceed 8 million files by 2025. Furthermore, according to the Ministry of Communication and Digital Affairs (Komdigi) of the Republic of Indonesia, the prevalence of deepfake content has risen sharply. Data from AI threat detection platforms indicate that the volume of deepfake content has increased by 550% since 2019, with 90% of such content being used for malicious purposes. This study employs a Convolutional Neural Network (CNN) with the EfficientNet-B0 architecture to classify real and deepfake faces using a dataset of Indonesian facial images. The model demonstrates strong and consistent performance in detecting deepfake faces, achieving 98% accuracy, an average F1-score above 0.96, and an AUC value in the range of 0.999–1.000. t-SNE visualization confirms clear separation between real and deepfake feature representations. Grad-CAM analysis indicates that decisions are mainly influenced by facial skin textures in the cheek and jaw regions, with minimal background contribution. These results support the development of automated deepfake detection systems and the enhancement of AI-based cyber security in Indonesia.*



**Keywords**: *generative AI, deepfake, deep learning, image classification, CNN*

# BAB I
# PENDAHULUAN

## 1.1. Latar Belakang Masalah

Bidang pengembangan citra sintetis berbasis Kecerdasan Buatan (*Artificial Intelligence*/AI) mengalami kemajuan yang signifikan dalam beberapa tahun terakhir. Pada tahap awal perkembangannya, teknologi generatif umumnya menghasilkan gambar dengan berbagai cacat visual yang mencolok sehingga relatif mudah dibedakan oleh pengamatan manusia. Namun, kondisi tersebut kini mengalami pergeseran karena model AI telah mampu menghasilkan gambar dengan tingkat presisi yang tinggi dan kualitas fotorealistik dalam waktu yang sangat singkat (Roose, 2022). Perkembangan AI yang berlangsung secara cepat telah mengantarkan kita pada suatu *fase* di mana gambar hasil generasi AI semakin menyerupai foto autentik dan sulit dibedakan secara visual (Bartos et al., 2023).

Wajah merupakan fitur yang paling khas dari manusia. Dengan pesatnya perkembangan teknologi sintesis wajah, risiko keamanan yang ditimbulkan oleh manipulasi wajah menjadi semakin signifikan. Wajah seseorang kini sering kali dapat dimanipulasi karena adanya berbagai algoritma yang berbasis teknologi *deep learning* (I. J. Goodfellow et al., 2014). Penerapan teknik *deep learning* (DL) generatif telah menghasilkan transformasi yang signifikan dalam ranah pemrosesan multimedia. Salah satu fenomena yang muncul sebagai konsekuensi dari perkembangan tersebut adalah *deepfake* (DF), yaitu teknologi yang memungkinkan penciptaan konten sintetis dengan memanfaatkan data citra dan video individu yang telah direkam sebelumnya (Patel et al., 2023).

*Deepfake* merupakan istilah yang berasal dari gabungan kata *deep learning* dan *fake*. Teknologi ini memungkinkan seseorang atau bahkan pengguna *machine learning* tanpa latar belakang teknis yang mendalam, untuk memanipulasi citra atau video melalui proses penggantian atau pertukaran konten. Hasil dari proses tersebut adalah media visual baru yang tampak sangat realistis, sehingga sulit dibedakan dari konten asli, baik oleh pengamatan manusia maupun oleh sistem komputasi (Malik et al., 2022). Istilah *deepfake* mulai dikenal luas pada akhir tahun 2017, ketika seorang pengguna *platform* Reddit dengan nama samaran "*deepfakes*"

mengungkapkan bahwa ia telah mengembangkan algoritma *machine learning* yang mampu menggantikan wajah selebritas ke dalam video pornografi. Seiring perkembangannya, penggunaan *deepfake* tidak hanya terbatas pada praktik pornografi palsu, tetapi juga dimanfaatkan untuk tujuan yang lebih berbahaya, seperti penyebaran berita bohong dan *hoax*, serta berbagai bentuk penipuan finansial yang dapat menimbulkan dampak sosial dan ekonomi yang signifikan (Tolosana et al., 2020).

Berbagai laporan internasional menunjukkan bahwa penggunaan dan penyebaran teknologi *deepfake* secara global terus mengalami peningkatan yang signifikan. Menurut laporan *Deepfake Statistics 2025 (AI Fraud Data & Trends)*, jumlah *file deepfake* secara global meningkat dari sekitar 5,5 juta pada tahun 2023 dan diproyeksikan mencapai lebih dari 8 juta *file* pada tahun 2025 (Khalil, 2025). Dari sisi jenis konten, artikel *Deepfake Statistics By Types, Fraud, Crime, Scams and Facts* melaporkan bahwa pada tahun 2025 konten *deepfake* didominasi oleh video dengan persentase sebesar 46%. Selanjutnya, *deepfake* berbentuk gambar mencakup 32%, sedangkan *deepfake* audio mencakup 22% dari keseluruhan konten palsu yang beredar. Kondisi ini semakin memprihatinkan karena sekitar 96% hingga 98% *deepfake* yang beredar di internet berupa konten intim yang dibuat tanpa persetujuan. Konten semacam ini secara langsung melanggar privasi individu dan dapat menimbulkan dampak psikologis serta sosial yang serius bagi para korbannya. Selain itu, maraknya *deepfake* menyebabkan perusahaan-perusahaan di berbagai negara mengalami kerugian rata-rata sekitar USD 500.000. Pada perusahaan berskala besar, kerugian yang ditimbulkan bahkan dapat mencapai USD 680.000 (Dey, 2025). Fenomena ini tidak hanya terbatas pada konteks internasional, tetapi juga telah merambah ke Indonesia di mana penggunaan media sosial yang masif mempercepat penyebaran konten *deepfake*.

Di Indonesia, tren penggunaan AI untuk membuat *deepfake* semakin meningkat. Hal ini sejalan dengan pertumbuhan jumlah pengguna internet yang pada tahun 2025 mencapai sekitar 229,4 juta jiwa. Tingkat penggunaan internet juga tergolong tinggi, yakni sekitar 80,66% dari total populasi (Unit Survey APJII, 2025). Menurut Kementerian Komunikasi dan Digital (Komdigi) Republik Indonesia, fenomena konten *deepfake* telah meningkat tajam. Data dari *platform*

deteksi ancaman AI menunjukkan bahwa jumlah konten *deepfake* meningkat sebesar 550% sejak 2019 dan 90% di antaranya digunakan untuk tujuan berbahaya (Komdigi, 2025). Teknologi *deepfake* mulai memainkan peran kunci sebagai alat baru untuk menipu dan menyebarkan hoaks. *Indonesia Anti-Scam Center* (IASC) mencatat lonjakan tajam laporan penipuan digital. Per Agustus 2025, laporan yang masuk tercatat sebanyak 225.281 dengan total kerugian sebesar 4.6 triliun rupiah (Safitri, 2025).

Berdasarkan hasil pemetaan yang dilakukan oleh Masyarakat Antifitnah Indonesia (Mafindo), pada rentang waktu 21 Oktober 2024 hingga 19 Oktober 2025 teridentifikasi sebanyak 1.593 kasus hoaks yang melibatkan penggunaan teknologi AI. Sebagian besar dari temuan tersebut berupa konten *deepfake* yang memanipulasi elemen wajah dan suara, sehingga berpotensi menyesatkan publik dan memperkuat penyebaran informasi palsu (Satrio, 2025). Banyaknya kasus pembuatan *deepfake* menyebabkan menurunnya tingkat kepercayaan masyarakat terhadap konten media digital. Masyarakat tidak lagi dapat sepenuhnya mempercayai gambar atau visual yang mereka lihat di berbagai *platform* digital (Malik et al., 2022).

Di Indonesia, tantangan permasalahan *deepfake* diperparah oleh kurangnya infrastruktur teknologi dan kesadaran masyarakat terhadap risiko *deepfake*. Data Indeks Masyarakat Digital Indonesia (IMDI) tahun 2025 menunjukkan bahwa tingkat literasi digital masyarakat Indonesia masih tergolong rendah dengan skor sebesar 49.28. Angka ini mengalami penurunan sebesar 8.97 poin dibandingkan tahun sebelumnya (Pudjianto, 2025). Oleh karena itu, masyarakat Indonesia rentan terhadap penyebaran konten *deepfake*, baik berupa video, gambar, maupun audio, yang dapat menimbulkan dampak negatif mulai dari penyebaran informasi palsu hingga pelanggaran privasi dan kerugian finansial.

Seiring semakin kaburnya batas antara konten nyata dan konten sintetis yang dihasilkan oleh AI, sangat penting untuk mengembangkan metode yang efektif dalam membedakan gambar yang dibuat oleh algoritma AI (seperti *Generative Adversarial Networks* (I. Goodfellow et al., 2020) atau *Diffusion Models* (Rombach et al., 2022)) dengan gambar yang berasal dari dunia nyata. Untuk mencapai hal tersebut, diperlukan teknik klasifikasi yang mampu mengenali pola-pola halus pada

citra yang sulit dideteksi oleh mata manusia. Klasifikasi dapat dilakukan dengan banyak metode, salah satunya yaitu menggunakan *Deep Learning* (Nugroho et al., 2020).

Salah satu metode deep learning yang efektif untuk mendeteksi *deepfake* adalah *Convolutional Neural Networks* (CNN) (Shad et al., 2021). *Convolutional Neural Network* (CNN) dirancang secara khusus untuk menangani tugas pengenalan serta klasifikasi citra. Arsitektur CNN terdiri atas beberapa lapisan (*layer*) yang berfungsi mengekstraksi fitur-fitur penting dari data visual, kemudian memanfaatkan informasi tersebut untuk menentukan kelas citra yang dihasilkan dalam bentuk skor klasifikasi (Nugroho et al., 2020). Arsitektur CNN yang efektif untuk *dataset* kecil atau terbatas adalah EfficientNet-B0 karena dirancang dengan pendekatan *compound scaling* yang menyeimbangkan kedalaman, lebar, dan resolusi jaringan secara optimal. Sebagai model *baseline* dalam keluarga EfficientNet, EfficientNet-B0 memiliki jumlah parameter yang relatif sedikit namun mampu mencapai akurasi yang kompetitif, sehingga mengurangi risiko *overfitting* pada *dataset* berukuran kecil (Tan & Le, 2019).

Beberapa penelitian terkait deteksi gambar *deepfake* menggunakan CNN dengan arsitektur EfficientNet-B0 telah dilakukan oleh (Agoncillo et al., 2024), (Şafak & Barışçı, 2024), (Adventino Gulo et al., 2025), dan (Tripathi, 2025). Dari uraian tersebut, telah banyak peneliti yang membahas tentang deteksi *deepfake* menggunakan CNN dengan arsitektur EfficientNet-B0. Namun demikian, penelitian tersebut masih terbatas pada *dataset* wajah dan gambar orang luar negeri. Belum ada penelitian yang menggunakan *dataset* gambar wajah orang Indonesia. Oleh karena itu, penelitian ini bertujuan melengkapi keterbatasan tersebut dengan menggunakan arsitektur CNN EfficientNet-B0 untuk klasifikasi wajah asli dan *deepfake* pada *dataset* wajah orang Indonesia. Kontribusi baru yang diberikan adalah pengembangan *dataset* spesifik wajah orang Indonesia, yang dapat memperkaya literatur deteksi *deepfake* dengan perspektif regional sekaligus memberikan dasar bagi penelitian dan pengembangan model yang lebih akurat untuk populasi lokal.

Secara teoretis, penelitian ini akan memberikan wawasan baru tentang efektivitas arsitektur CNN dalam menangani variasi etnis dan memperkaya teori

pembelajaran mesin untuk aplikasi forensik digital. Secara praktis, hasil penelitian ini dapat dimanfaatkan oleh sektor keamanan siber. Misalnya, *platform* media sosial di Indonesia dapat menggunakannya untuk mengurangi risiko penyebaran misinformasi. Selain itu, pemerintah dapat memanfaatkan hasil ini untuk merancang strategi keamanan digital yang lebih efektif.

## 1.2. Rumusan Masalah

Adapun rumusan masalah dari penelitian ini yaitu:

1. Bagaimana implementasi metode CNN dengan arsitektur EfficientNet-B0 dalam mengklasifikasikan wajah asli dan wajah *deepfake* (*AI-generated*) pada *dataset* wajah orang Indonesia?
2. Bagaimana tingkat akurasi dan kinerja model CNN dengan arsitektur EfficientNet-B0 dalam membedakan wajah asli dan wajah *deepfake* (*AI-generated*) pada *dataset* wajah orang Indonesia?

## 1.3. Batasan Masalah

Batasan masalah dari penelitian ini adalah sebagai berikut:

1. Data *deepfake* yang dianalisis dalam penelitian ini dibatasi hanya pada *deepfake* berupa citra gambar wajah, tidak mencakup video, audio, atau bentuk manipulasi lainnya.
2. Penelitian ini hanya berfokus pada wajah dan tidak mempertimbangkan ornamen atau atribut di luar area wajah, seperti hijab, topi, dan aksesoris lainnya.
3. Citra wajah yang digunakan dalam penelitian ini terbatas pada wajah orang Indonesia, baik untuk citra wajah asli maupun citra wajah *deepfake* (*AI-generated*).
4. Analisis dan pengolahan citra dalam penelitian ini dilakukan menggunakan *platform* Google Colaboratory (Google Colab) dengan penerapan arsitektur CNN EfficientNet-B0.

## 1.4. Tujuan Penelitian

Adapun tujuan penelitian yang ingin dicapai adalah sebagai berikut:

1. Mengetahui implementasi metode CNN dengan arsitektur EfficientNet-B0 dalam mengklasifikasikan wajah asli dan wajah *deepfake* (*AI-generated*) pada *dataset* wajah orang Indonesia.

2. Mengetahui tingkat akurasi dan kinerja model CNN dengan arsitektur EfficientNet-B0 dalam membedakan wajah asli dan wajah *deepfake* (*AI-generated*) pada *dataset* wajah orang Indonesia.

## 1.5. Manfaat Penelitian

Adapun manfaat penelitian yang ingin dicapai adalah sebagai berikut:

1. Memberikan pemahaman dan wawasan mengenai penerapan teknologi deteksi *deepfake* pada citra wajah orang Indonesia.
2. Mengevaluasi kinerja dan tingkat akurasi model CNN dengan arsitektur EfficientNet-B0 dalam mengklasifikasikan wajah asli dan wajah *deepfake* (*AI-generated*), sehingga berkontribusi terhadap pengembangan metode *deep learning* dalam bidang *computer vision* dan keamanan digital.
3. Menyediakan referensi ilmiah bagi penelitian selanjutnya yang berkaitan dengan deteksi *deepfake*, khususnya yang menggunakan *dataset* lokal Indonesia, sehingga dapat memperkaya literatur dan mendorong pengembangan riset yang lebih kontekstual dan representatif.
4. Memberikan informasi yang bermanfaat bagi pemerintah, institusi pendidikan, dan *platform* digital dalam merancang strategi literasi digital serta kebijakan keamanan siber guna mengantisipasi penyebaran konten visual palsu berbasis AI.

# BAB II
# TINJAUAN PUSTAKA

Tinjauan pustaka dalam penelitian ini mengacu pada berbagai studi terdahulu yang telah dilakukan di bidang deteksi *deepfake* dan klasifikasi citra wajah berbasis *deep learning*. Tinjauan pustaka ini berfungsi sebagai landasan teoretis dan metodologis terkait penggunaan CNN dalam klasifikasi wajah asli dan *deepfake*. Selain itu, tinjauan ini juga membantu mengidentifikasi celah penelitian yang masih terbuka, terutama dalam pemanfaatan *dataset* wajah orang Indonesia. Dengan demikian, penelitian ini diharapkan dapat memberikan kontribusi yang bermakna dan berbeda dari studi-studi sebelumnya sekaligus memperkuat keterkaitan dengan basis pengetahuan yang telah ada dalam bidang *computer vision* dan keamanan digital.

## 2.1. Penelitian Terdahulu

Penelitian terdahulu merupakan bagian penting dalam kajian ilmiah yang bertujuan untuk meninjau, membandingkan, dan mengevaluasi berbagai studi yang telah dilakukan sebelumnya terkait deteksi *deepfake* dan klasifikasi citra wajah berbasis *deep learning*. Melalui analisis terhadap penelitian-penelitian tersebut, dapat diidentifikasi pendekatan, metode, arsitektur model, serta jenis *dataset* yang digunakan, termasuk kelebihan dan keterbatasan masing-masing penelitian. Selain itu, kajian ini juga berperan dalam menemukan *research gap* yang masih terbuka, khususnya dalam konteks penggunaan *dataset* wajah orang Indonesia yang masih relatif terbatas. Oleh karena itu, penelitian ini disusun dengan mempertimbangkan hasil-hasil penelitian sebelumnya guna mengembangkan pendekatan yang lebih relevan, adaptif, dan kontekstual terhadap permasalahan yang diangkat.

**Tabel 2.1.** Tabel Penelitian Sebelumnya

| Tahun | Nama | Judul | Data | Hasil Penelitian |
|---|---|---|---|---|
| 2023 | Samiya Afzal Minhas, Saad Mushtaq, Ali Javed | *EfficientNetB0 Ensemble Model for Unified Deepfakes Detection* | Penelitian ini menggunakan dua *dataset* untuk mengevaluasi model, yaitu FaceForensics++ dan Celeb-DF. FaceForensics++ terdiri dari 1.000 | Penelitian ini menunjukkan bahwa EfficientNet-B0 mampu memberikan akurasi deteksi *deepfake* yang sangat tinggi pada *dataset* berskala besar. Pengujian |

| Tahun | Nama | Judul | Data | Hasil Penelitian |
|---|---|---|---|---|
| | | | video wajah asli dan 4.000 video hasil manipulasi (*deepfake*) dari beberapa metode, sehingga total berjumlah 5.000 video. Celeb-DF terdiri dari 408 video wajah asli dan 795 video *deepfake* yang dihasilkan dari video selebritas, dengan total 1.203 video. | pada *dataset* FaceForensics++ dan CelebDF menghasilkan akurasi masing-masing sebesar 99,7% dan 96,09%. Hasil tersebut menegaskan bahwa EfficientNet-B0 merupakan arsitektur yang efektif dan andal untuk deteksi *deepfake*. |
| 2024 | Rafael Alden F. Agoncillo, James Kenneth M. Kiunisala, Raymund M. | *Enhancement of Deepfake Detection Framework Integrating Efficient NetB0, Graph Attention Networks, and Gated Recurrent Units* | Data yang digunakan adalah Celeb-DF V2. *Dataset* ini terdiri 590 video asli dan 5.639 video *deepfake* yang dimanipulasi menggunakan teknik seperti *FaceSwap* dan *DFaker*. Seluruh video pada *dataset* ini berasal dari 59 selebriti. | Penelitian ini menggunakan EfficientNet-B0 sebagai ekstraksi fitur yang dikombinasikan dengan *Graph Attention Networks* dan GRU untuk deteksi *deepfake*. Hasil evaluasi menunjukkan akurasi 80.60% dengan nilai loss 0.28. Model ini mampu mengungguli Mesonet, ResNet-50, VGG-19, dan Xception. Hasil ini menunjukkan efektivitas EfficientNet-B0 dalam mendukung deteksi *deepfake* yang andal. |
| 2024 | M. Umadevi, S. Barghav Krishna, N. Sai Kumar | *Deep Fake Face Detection using Efficient Convolutional Neural Networks* | *Dataset* berasal dari Kaggle, yang terdiri dari 140.000 gambar citra wajah. | Hasil eksperimen menunjukkan bahwa EfficientNet-B0 dan MobileNet mencapai akurasi tertinggi, masing-masing sebesar 98% dalam 15 epoch. Kinerja tersebut melampaui model CNN yang |

| Tahun | Nama | Judul | Data | Hasil Penelitian |
|---|---|---|---|---|
| | | | | diusulkan (96%) dan DenseNet121 (97%). Temuan ini menegaskan efektivitas EfficientNet-B0 sebagai arsitektur CNN yang unggul untuk deteksi *deepfake* berbasis citra. |
| 2024 | Emre Şafak, Necaattin Barışçı | *Detection of fake face images using lightweight convolutional neural networks with stacking ensemble learning method* | Data yang digunakan adalah *Flickr-Faces-HQ* (FFHQ). *Dataset* terdiri dari 70.000 citra wajah asli dan 70.000 citra wajah palsu hasil StyleGAN2 untuk mendeteksi manipulasi wajah. | Beberapa arsitektur CNN yaitu MobileNet, MobileNetV2, EfficientNet-B0, dan NASNetMobile, dievaluasi menggunakan *transfer learning*. Hasil awal menunjukkan bahwa EfficientNet-B0 mencapai akurasi tertinggi sebesar 93.64% dibandingkan model lainnya. Setelah dilakukan modifikasi arsitektur, EfficientNet-B0 berhasil meningkatkan akurasi hingga 95.48%. Hal ini menegaskan keunggulan EfficientNet-B0 sebagai model dasar deteksi wajah *deepfake*. |
| 2024 | Rimjhim Padam Singh, Nichenametla Hima Sree, Koti Leela Sai Praneeth Reddy, Kandukuri Jashwanth | *Convergence of Deep Learning and Forensic Methodologies Using Self-attention Integrated EfficientNet* | Data yang digunakan adalah *dataset* Celeb-DF. *Dataset* ini terdiri atas 408 video asli yang dikumpulkan dari YouTube dan 795 video *deepfake* yang disintesis dari | Penelitian ini mengusulkan model deteksi *deepfake* berbasis EfficientNet-B0 yang dilakukan *fine-tuning* dan dipadukan dengan mekanisme *self-* |

| Tahun | Nama | Judul | Data | Hasil Penelitian |
|---|---|---|---|---|
| | | *Model for Deep Fake Detection* | video-video asli tersebut. Seluruh video pada *dataset* ini berasal dari 59 selebriti. | *attention*. Evaluasi dilakukan dengan membandingkan beberapa model CNN lain, yaitu XceptionNet, NasNetLarge, InceptionV3, ConvNeXt, dan EfficientNet-B0 standar. Hasil eksperimen menunjukkan bahwa EfficientNet-B0 dengan *self-attention* memberikan performa terbaik dengan akurasi sebesar 97,67%. Temuan ini menegaskan keunggulan EfficientNet-B0 dibandingkan arsitektur CNN lainnya dalam deteksi gambar *deepfake*. |
| 2025 | Steven Adventino Gulo, Ayu Amelia Pertiwi, Sarah Putri Syaifullah Nasution, Hermawan Syahputra | *Deteksi Deepfake Dalam Citra Menggunakan Convolutional Neural Network (CNN)* | *Dataset* diambil dari DeepFakeDetection di situs HuggingFace yang terdiri dari 10.000 gambar asli dan 10.000 gambar *fake* | Model CNN dengan arsitektur EfficientNet-B0 yang dikombinasikan dengan ekstraksi *facial landmark* menggunakan OpenCV menghasilkan performa klasifikasi yang tinggi. Hasil evaluasi menunjukkan nilai akurasi sebesar 91% berdasarkan *confusion matrix*. Temuan ini menunjukkan bahwa EfficientNet-B0 efektif digunakan untuk deteksi wajah |

| Tahun | Nama | Judul | Data | Hasil Penelitian |
|---|---|---|---|---|
| | | | | *deepfake* berbasis citra. |
| 2025 | Yashoda Chouhan, Chetankumar Chudasama, Deepak Kumar Verma | *DeDER: Detecting Deepfakes with EfficientNetB0 and ResNet50 on Celeb-DF* | Data yang digunakan adalah *dataset* Celeb-DF. *Dataset* ini terdiri atas 408 video asli yang dikumpulkan dari YouTube dan 795 video *deepfake* yang disintesis dari video-video asli tersebut. Seluruh video pada *dataset* ini berasal dari 59 selebriti. | Penelitian ini mengevaluasi kinerja EfficientNet-B0 dan ResNet50 dalam mendeteksi *deepfake*. Hasil eksperimen menunjukkan bahwa EfficientNet-B0 mencapai akurasi 98.17%, lebih tinggi dibandingkan ResNet50 dengan akurasi 98,03%. Temuan ini menunjukkan bahwa EfficientNet-B0 memiliki performa yang sangat baik sebagai model tunggal dalam sistem deteksi *deepfake*. |
| 2025 | Shashank Mani Tripathi, Priyanshu Srivastava, Devansh Yadav, Prachi Verma | *Classification of AI-Generated, Deepfake and Real Images Using CNN* | *Dataset* berasal dari HuggingFace dan berisi 9.999 citra wajah RGB yang diklasifikasikan ke dalam tiga kelas, yaitu wajah hasil generasi AI, *deepfake*, dan wajah asli. Karena hanya tersedia data pelatihan, dilakukan pembagian *stratified* 80:20 menjadi 7.999 data latih dan 2.000 data validasi. | Penelitian ini mengusulkan kerangka klasifikasi tiga kelas (citra asli, *deepfake*, dan *AI-generated*) dengan menggunakan EfficientNet-B0 sebagai *backbone* utama. Hasil evaluasi menunjukkan performa yang sangat tinggi dengan akurasi validasi sebesar 99,8% serta nilai ROC–AUC yang kuat. Temuan ini membuktikan bahwa EfficientNet-B0 mampu menangani klasifikasi citra manipulasi modern secara efektif. |

# BAB III
# LANDASAN TEORI

## 3.1. *Deepfake*

*Deepfake* merupakan istilah yang dibentuk dari kata *deep learning* dan *fake*, yang menggambarkan pemanfaatan metode pembelajaran mendalam untuk memanipulasi atau menghasilkan media digital secara sintetis . Teknologi ini dapat diterapkan pada berbagai jenis media, termasuk gambar, video, dan audio, baik dengan memodifikasi konten yang sudah ada maupun dengan menciptakan konten baru sepenuhnya (Altuncu et al., 2024). Teknologi *deepfake* dikembangkan dengan memanfaatkan jaringan saraf berbasis *deep learning* yang dilatih menggunakan data visual dan audio dalam skala besar. Model ini mempelajari karakteristik unik seseorang, seperti ekspresi wajah, gerakan, serta pola suara, sehingga mampu merekonstruksi tampilan dan perilaku yang menyerupai individu tersebut secara realistis (Westerlund, 2019).

Manipulasi konten digital bukanlah fenomena baru, namun kemunculan teknologi *deepfake* membawa kemampuan tersebut ke tingkat yang jauh lebih kompleks dan realistis (B. U. Mahmud & Sharmin, 2021). Sejak kemunculannya ke ranah publik pada tahun 2017, teknologi *deepfake* menarik perhatian luas akibat penyalahgunaannya, khususnya pada konten sensitif yang melibatkan figur publik. *Deepfake* menargetkan *platform* media sosial, yaitu lingkungan digital di mana teori konspirasi, rumor, dan misinformasi dapat menyebar dengan mudah (Westerlund, 2019). *Deepfake* yang memanfaatkan algoritma *machine learning* dan kecerdasan buatan mampu memodifikasi konten visual dan audio asli menjadi representasi palsu yang sulit dibedakan dari konten autentik (B. U. Mahmud & Sharmin, 2021). Hal ini menyebabkan banyak pengguna media digital secara tidak sadar mempercayai konten *deepfake* sebagai informasi yang sah, sehingga menimbulkan tantangan serius dalam aspek keamanan informasi dan kepercayaan publik (Westerlund, 2019).

Seiring dengan pesatnya perkembangan teknologi *deep learning*, proses pembuatan konten palsu menjadi semakin cepat dan mudah dilakukan, bahkan oleh individu dengan kemampuan komputasi yang relatif terbatas. Teknologi ini telah

dimanfaatkan dalam berbagai bentuk penyalahgunaan, seperti pembuatan konten pornografi palsu yang melibatkan figur publik, penyebaran disinformasi, pemalsuan suara tokoh penting, hingga tindak kejahatan finansial. Tingginya tingkat realisme pada konten yang dihasilkan menyebabkan konten palsu tersebut tampak kredibel dan sulit dibedakan dari media asli. Kondisi ini secara signifikan meningkatkan risiko penyebaran konten palsu di ruang digital, sehingga menegaskan urgensi penelitian terkait pengembangan metode deteksi wajah *deepfake* secara otomatis dan andal (B. U. Mahmud & Sharmin, 2021).

### 3.1.1. *Deepfake* Citra Wajah

*Deepfake* gambar atau citra merupakan istilah yang merujuk pada teknik manipulasi maupun pembuatan citra digital dengan memanfaatkan kecerdasan buatan, khususnya pendekatan *deep learning*. Teknologi ini mampu menghasilkan visual yang sangat realistis, meskipun bersifat palsu dan tidak merepresentasikan kondisi yang sebenarnya. *Deepfake* memanfaatkan model-model generatif seperti *Generative Adversarial Networks* (GAN) atau arsitektur lain untuk mempelajari pola dan distribusi data citra nyata, kemudian menciptakan citra baru atau memodifikasi citra asli sehingga tampak otentik. Teknologi ini memungkinkan konten visual direkayasa dalam konteks yang sulit dibedakan oleh persepsi manusia biasa dan dapat digunakan baik untuk tujuan kreatif maupun disinformasi (Shahzad et al., 2022).

Wajah menjadi objek utama berbagai teknik manipulasi digital karena telah banyak diteliti dalam bidang *computer vision* sebagai dasar pengembangan teknologi manipulasi citra, sekaligus memiliki peran sentral dalam komunikasi manusia (Rossler et al., 2019). Wajah manusia memiliki peran yang sangat penting dalam komunikasi digital karena mengandung informasi identitas, ekspresi emosi, dan pesan nonverbal yang kuat. Sebagai ciri biometrik utama, wajah sering digunakan sebagai representasi identitas seseorang dalam berbagai konteks. Oleh karena itu, kemunculan alat berbasis kecerdasan buatan yang mampu menghasilkan wajah manusia secara realistis dari individu yang sebenarnya tidak pernah ada atau memodifikasi atribut wajah telah menimbulkan kekhawatiran yang signifikan (Verdoliva, 2020).

### 3.1.2. Jenis Manipulasi Wajah

Tolosana dkk. (2020) mengelompokkan manipulasi wajah (*facial manipulation*) ke dalam empat kategori utama berdasarkan tingkat manipulasi yang diterapkan pada konten visual.

1. *Entire Face Synthesis*

   *Entire face synthesis* merupakan jenis manipulasi wajah yang menghasilkan citra wajah manusia baru secara keseluruhan tanpa merepresentasikan identitas individu yang nyata. Meskipun memiliki berbagai manfaat, seperti pembuatan karakter digital pada industri permainan (*video game*), perfilman, serta pemodelan tiga dimensi, teknologi ini juga berpotensi disalahgunakan. Penyalahgunaan tersebut antara lain meliputi pembuatan identitas palsu di media sosial, pembuatan profil yang tampak autentik untuk tujuan penipuan, serta penyebaran informasi yang menyesatkan.

2. *Identity Swap*

   *Identity swap* merupakan teknik manipulasi wajah yang bertujuan menggantikan identitas seseorang dengan identitas individu lain. Manipulasi ini dapat dilakukan menggunakan metode grafika komputer konvensional maupun pendekatan berbasis *deep learning*. Dalam prosesnya, teknik *identity swap* mempertahankan sebagian besar karakteristik visual asli, tetapi mengganti identitas individu yang ditampilkan sehingga tampak seolah-olah orang lain berada dalam konten tersebut. Teknologi ini memiliki berbagai manfaat, misalnya untuk menghasilkan efek visual dalam industri perfilman dan hiburan. Di sisi lain, teknologi ini juga berpotensi disalahgunakan untuk membuat video pornografi palsu, menyebarkan hoaks dan disinformasi, serta melakukan berbagai bentuk penipuan.

3. *Attribute Manipulation*

   *Attribute manipulation* atau *face editing* merupakan teknik manipulasi wajah yang mengubah atribut tertentu tanpa mengubah identitas individu secara keseluruhan. Manipulasi ini dapat dilakukan pada berbagai atribut visual, seperti warna rambut, warna kulit, usia, jenis kelamin,

penambahan kacamata, janggut, maupun karakteristik wajah lainnya. Apabila tidak disalahgunakan, teknologi ini memiliki berbagai manfaat dalam kehidupan sehari-hari. Salah satu penerapannya adalah simulasi virtual untuk mencoba produk, seperti kosmetik, gaya rambut, dan aksesori, sebelum digunakan secara nyata.

4. *Face Reenactment*

   *Expression swap* atau *face reenactment* merupakan teknik manipulasi wajah yang mengubah ekspresi wajah seseorang tanpa mengubah identitasnya. Pada teknik ini, ekspresi wajah dari individu lain ditransfer ke wajah target seolah-olah individu tersebut benar-benar menampilkan ekspresi tersebut. Beberapa metode yang umum digunakan dalam *face reenactment* antara lain *Face2Face* dan *NeuralTextures*. Teknologi ini banyak dimanfaatkan dalam industri hiburan, animasi digital, dan produksi efek visual. Namun, teknologi ini juga berpotensi disalahgunakan untuk tujuan negatif yang dapat memicu penyebaran disinformasi dan menurunkan kepercayaan masyarakat terhadap konten digital.

### 3.2. *Artificial Intelligence* (AI)

*Artificial Intelligence* (AI) merupakan cabang ilmu komputer yang berfokus pada pengembangan sistem yang mampu meniru kecerdasan dan perilaku manusia untuk membantu menyelesaikan berbagai pekerjaan secara lebih efektif. Tujuan utama AI adalah mereplikasi kemampuan intelektual manusia sehingga komputer dapat menyelesaikan tugas-tugas yang umumnya memerlukan kecerdasan manusia. Selain itu, AI bertujuan membangun sistem yang mampu menangani permasalahan yang bersifat kompleks dan membutuhkan pengetahuan (*knowledge-intensive tasks*). AI juga dikembangkan agar mampu belajar dari data dan pengalaman secara mandiri sehingga kinerjanya dapat terus meningkat tanpa memerlukan pemrograman ulang secara eksplisit (Ghosh & Thirugnanam, 2019). Konsep AI pertama kali diperkenalkan oleh McCarthy pada tahun 1956 yang mendefinisikan AI sebagai ilmu dan rekayasa untuk membuat mesin cerdas, khususnya program komputer yang mampu menampilkan perilaku cerdas seperti manusia (McCarthy, 2007).

AI tidak hanya berfokus pada pemahaman mengenai kecerdasan, tetapi juga pada pengembangan sistem atau mesin yang mampu bertindak secara cerdas dalam berbagai situasi. Sistem AI dirancang agar dapat menentukan tindakan yang efektif dan aman ketika menghadapi kondisi yang baru atau belum pernah ditemui sebelumnya. Dengan kemampuan tersebut, AI dapat membantu menyelesaikan berbagai permasalahan yang umumnya memerlukan kecerdasan manusia. AI mencakup berbagai subbidang ilmu yang saling berkaitan, mulai dari pembelajaran (*learning*), penalaran (*reasoning*), persepsi (*perception*), pemrosesan bahasa alami (*natural language processing*), robotika, hingga visi komputer (*computer vision*). Masing-masing subbidang dikembangkan untuk memungkinkan sistem AI memahami lingkungan, memproses informasi, serta mengambil keputusan sesuai dengan tujuan yang telah ditentukan (Russell & Norvig, 2021).

### 3.2.1. Jenis *Artificial Intelligence* (AI)

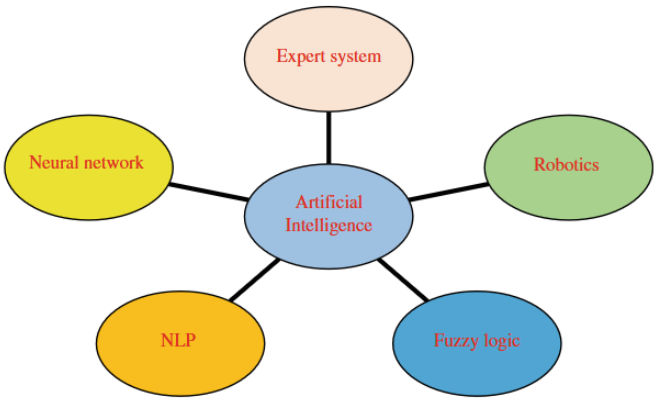
Berdasarkan tingkat kemampuannya, AI dapat diklasifikasikan menjadi *Weak AI (Narrow AI)*, *General AI*, dan *Strong AI*. *Weak AI* merupakan jenis AI yang paling banyak digunakan saat ini, di mana sistem dirancang untuk menjalankan tugas tertentu tanpa memiliki kemampuan berpikir atau kesadaran sebagaimana manusia. *General AI* merupakan konsep AI yang diharapkan mampu melakukan berbagai aktivitas intelektual seperti manusia, namun hingga saat ini belum berhasil direalisasikan. Sementara itu, *Strong AI* merupakan konsep AI yang diharapkan memiliki kemampuan yang melampaui manusia dalam berpikir dan menyelesaikan berbagai permasalahan, tetapi hingga saat ini belum memiliki implementasi nyata (Ghosh & Thirugnanam, 2019; Singh, 2017).

Selain berdasarkan kemampuan, AI juga dapat diklasifikasikan berdasarkan fungsionalitasnya menjadi empat jenis, yaitu *Reactive Machines*, *Limited Memory*, *Theory of Mind*, dan *Self-Awareness*. *Reactive Machines* merupakan sistem AI yang hanya merespons kondisi saat ini tanpa memiliki kemampuan menyimpan pengalaman sebelumnya. *Limited Memory* memiliki kemampuan memanfaatkan data atau pengalaman masa lalu dalam jangka waktu tertentu untuk mendukung proses pengambilan keputusan. Sementara itu, *Theory of Mind* merupakan konsep AI yang diharapkan mampu memahami kondisi mental manusia, seperti emosi, keyakinan, niat, harapan, dan keinginan. Adapun *Self-Awareness* merupakan

konsep AI yang diharapkan memiliki kesadaran diri sebagaimana manusia. Kedua jenis terakhir tersebut hingga saat ini belum berhasil direalisasikan sehingga masih menjadi arah pengembangan teknologi AI di masa mendatang (Ghosh & Thirugnanam, 2019). Dalam konteks penelitian ini, teknologi *deepfake* dan metode Convolutional Neural Network (CNN) termasuk ke dalam kategori *Weak AI* dan *Limited Memory*, karena sistem hanya dilatih untuk melakukan klasifikasi citra berdasarkan data yang tersedia tanpa memiliki pemahaman atau kesadaran layaknya manusia.

### 3.2.2. Domain *Artificial Intelligence* (AI)

AI memiliki beberapa domain utama yang merepresentasikan bidang penerapan dan pendekatan teknologi kecerdasan buatan, yaitu jaringan saraf tiruan (*neural networks*), robotika, sistem pakar (*expert systems*), sistem logika *fuzzy*, dan pemrosesan bahasa alami (*natural language processing/NLP*). Setiap domain memiliki karakteristik dan tujuan yang berbeda sesuai dengan permasalahan yang ingin diselesaikan.




**Gambar 3.1.** Berbagai Domain AI (Ghosh & Thirugnanam, 2019)

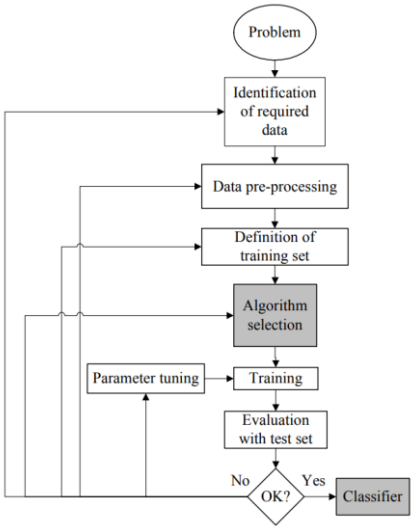
*Neural networks* merupakan domain AI yang meniru cara kerja sistem saraf manusia melalui lapisan neuron untuk mempelajari pola dan hubungan kompleks dalam data. Domain ini menjadi dasar pengembangan *machine learning* dan *deep learning*, serta banyak diterapkan pada pengenalan pola, seperti deteksi wajah dan klasifikasi citra. Robotika berfokus pada pengembangan mesin cerdas dalam bentuk robot yang mampu menjalankan instruksi manusia. Sistem pakar dirancang untuk meniru proses pengambilan keputusan seorang ahli berdasarkan basis pengetahuan tertentu, sedangkan logika *fuzzy* meniru cara berpikir manusia dalam kondisi

ketidakpastian dengan mempertimbangkan nilai di antara benar dan salah. Sementara itu, NLP berfokus pada interaksi antara komputer dan bahasa manusia, seperti pada sistem penerjemah dan pemeriksa ejaan otomatis (Ghosh & Thirugnanam, 2019). Dalam penelitian ini, metode yang digunakan termasuk dalam domain *neural networks* untuk melakukan klasifikasi wajah asli dan wajah *deepfake*.

### 3.3. *Machine Learning* (ML)

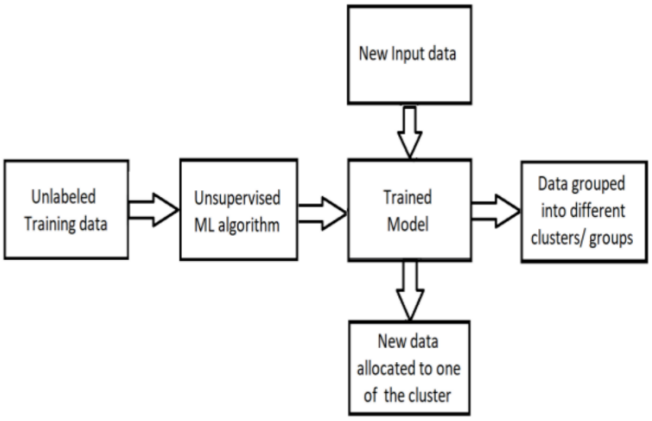
*Machine Learning* merupakan cabang dari kecerdasan buatan (AI) yang memungkinkan sistem komputer untuk belajar dari data dan pengalaman sebelumnya tanpa harus diprogram secara eksplisit (Murdiawati et al., 2026). Dalam pendekatan ini, model dibangun melalui proses pelatihan (*training*) menggunakan data sehingga mampu mempelajari hubungan antara masukan dan keluaran untuk menyelesaikan tugas tertentu. Berbeda dengan pendekatan rekayasa konvensional yang mengandalkan perancangan aturan secara manual oleh pakar, *machine learning* memanfaatkan data dalam jumlah besar untuk melatih model melalui proses optimasi sehingga solusi yang dihasilkan diperoleh dari hasil pembelajaran terhadap data (Simeone, 2018).

Algoritma *machine learning* secara umum diklasifikasikan menjadi tiga kategori, yaitu *supervised learning* (SL), *unsupervised learning* (UL), dan *semi-supervised learning* (SSL). Perbedaan utama ketiga pendekatan tersebut terletak pada penggunaan label pada data pelatihan, yaitu apakah setiap sampel data telah memiliki label atau belum (Yan & Wang, 2022). Dalam *supervised learning*, model dilatih menggunakan data berlabel sehingga sistem dapat mempelajari hubungan antara fitur *input* dan target *output*. Pendekatan ini banyak digunakan pada permasalahan klasifikasi dan regresi, termasuk klasifikasi citra di mana label kelas telah diketahui sebelumnya (Bishop, 2006). Penelitian deteksi wajah *deepfake* termasuk ke dalam kategori *supervised learning* karena model dilatih untuk membedakan antara wajah asli dan wajah hasil manipulasi berdasarkan data yang telah diberi label. Alur kerja algoritma *supervised learning* dapat dilihat pada Gambar 3.2.




**Gambar 3.2.** Proses *Supervised Learning* (Kotsiantis, 2007)

Berbeda dengan *supervised learning*, *unsupervised learning* bertujuan untuk menemukan pola atau struktur yang menarik dalam data tanpa menggunakan label kelas. Pendekatan ini sering diterapkan pada tugas pengelompokan, penemuan faktor laten, dan reduksi dimensi. *Unsupervised learning* berperan penting dalam proses penemuan pengetahuan (*knowledge discovery*) serta pembelajaran representasi dari data (Murphy, 2012). Alur kerja algoritma *unsupervised learning* dapat dilihat pada Gambar 3.3. Sementara itu, *semi-supervised learning* merupakan kombinasi dari kedua pendekatan tersebut (Engelen & Hoos, 2020). Pendekatan ini memanfaatkan data berlabel (*labeled data*) dan data tidak berlabel (*unlabeled data*) secara bersamaan untuk membangun fungsi atau model klasifikasi yang optimal (Nasteski, 2017).




**Gambar 3.3.** *Unsupervised Learning Algorithm* (Pandey et al., 2019)

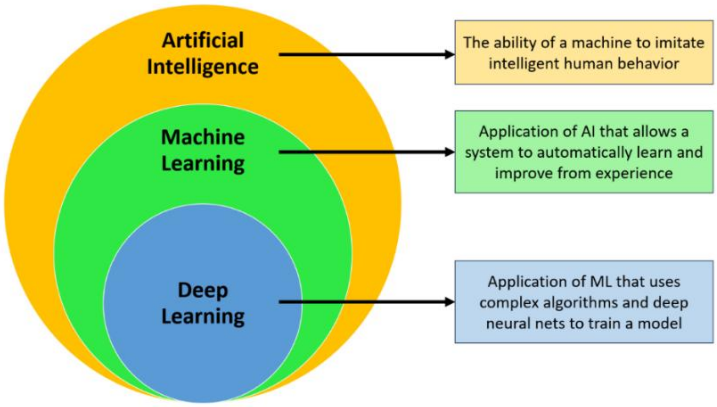
*Machine learning* menawarkan alternatif yang lebih efisien dibandingkan pendekatan rekayasa konvensional, terutama ketika biaya dan waktu

pengembangan menjadi pertimbangan utama atau ketika permasalahan yang dihadapi terlalu kompleks untuk dimodelkan secara menyeluruh (Simeone, 2018). Salah satu metode dalam *machine learning* adalah klasifikasi, yaitu teknik yang bertujuan mengelompokkan atau memprediksi data ke dalam kelas tertentu berdasarkan karakteristik yang dimilikinya (Pahlawan et al., 2024). Dalam penelitian ini, metode klasifikasi pada *machine learning* menjadi dasar untuk mengembangkan sistem yang mampu membedakan citra wajah asli dan citra wajah *deepfake* berdasarkan karakteristik yang dipelajari dari data.

### 3.4. *Deep Learning* (DL)

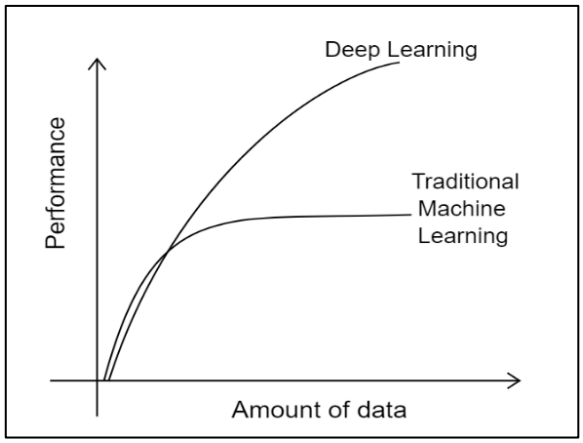
*Machine learning* dan *deep learning* merupakan metode dalam bidang kecerdasan buatan (AI) yang berkontribusi terhadap otomatisasi berbagai tugas intelektual yang sebelumnya dilakukan oleh manusia (Kufel et al., 2023). Seiring dengan perkembangan *machine learning*, berkembang pula pendekatan *deep learning* sebagai salah satu bentuk *representation learning* yang mampu mempelajari representasi data secara bertingkat sehingga memudahkan proses ekstraksi fitur secara otomatis dalam membangun model klasifikasi maupun prediksi (Alom et al., 2019).

Menurut LeCun et al. (2015), *deep learning* merupakan salah satu pendekatan *machine learning* yang menggunakan jaringan saraf tiruan dengan banyak lapisan pemrosesan untuk mempelajari representasi data pada berbagai tingkat abstraksi secara hierarkis. Istilah *deep* mengacu pada banyaknya lapisan pemrosesan nonlinier yang memungkinkan model mempelajari representasi fitur secara otomatis dari data mentah. Melalui representasi bertingkat tersebut, lapisan awal mempelajari fitur-fitur dasar, sedangkan lapisan yang lebih dalam menangkap pola yang semakin abstrak dan kompleks. Gambaran hubungan antara *artificial intelligence*, *machine learning*, dan *deep learning* dapat dilihat pada Gambar 3.4. Keunggulan *deep learning* dalam mempelajari pola kompleks dari data berskala besar menjadikannya pendekatan yang sangat relevan untuk aplikasi pengolahan citra. Dalam konteks penelitian ini, kemampuan tersebut dimanfaatkan untuk mendeteksi perbedaan halus antara wajah asli dan wajah *deepfake*.




**Gambar 3.4.** Hubungan *DL, ML,* dan *AI* (Raihan, 2023)

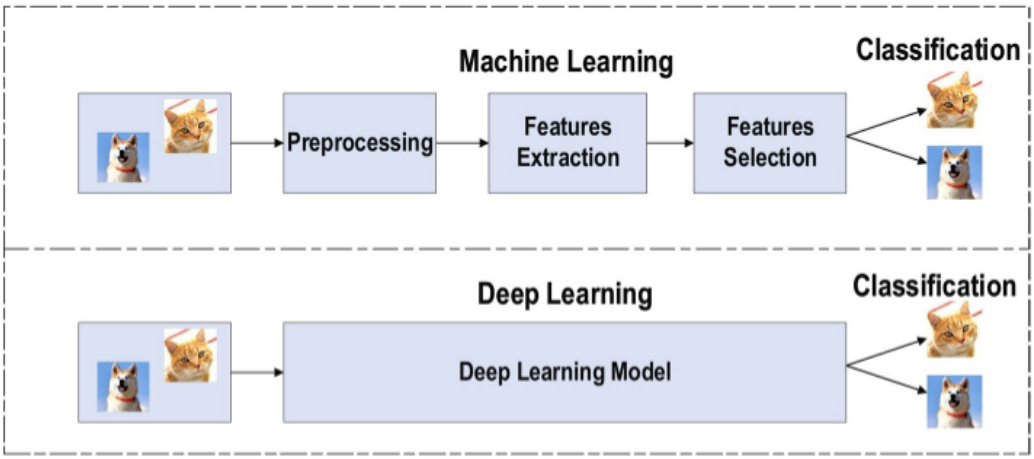
Perkembangan *deep learning* didukung oleh ketersediaan data dalam jumlah besar yang memungkinkan model mempelajari representasi yang semakin kompleks melalui proses pelatihan jaringan saraf berukuran besar (Alom et al., 2019). Kinerja pengklasifikasi berbasis *deep learning* menunjukkan peningkatan yang signifikan seiring dengan bertambahnya jumlah data, terutama jika dibandingkan dengan metode pembelajaran tradisional. Gambar 3.5 memperlihatkan perbandingan kinerja antara algoritma *machine learning* konvensional dan algoritma *deep learning*. Pada metode *machine learning* tradisional, peningkatan kinerja cenderung berhenti dan menjadi stabil setelah mencapai ambang tertentu dari jumlah data pelatihan. Sebaliknya, metode *deep learning* terus menunjukkan peningkatan kinerja seiring dengan bertambahnya volume data yang digunakan dalam proses pelatihan (Mathew et al., 2021).




**Gambar 3.5.** Kinerja ML dan DL (Mathew et al., 2021)

Berbeda dengan metode *machine learning* konvensional yang bergantung pada proses *feature engineering* secara manual, *deep learning* mampu melakukan ekstraksi fitur secara otomatis melalui arsitektur representasi data bertingkat (*multi-*

*layer data representation*). Lapisan awal mempelajari fitur tingkat rendah, sedangkan lapisan yang lebih dalam mengekstraksi fitur tingkat tinggi yang lebih diskriminatif. Kemampuan tersebut menjadikan *deep learning* memperoleh kinerja yang sangat baik pada berbagai aplikasi, seperti pengolahan citra, audio, dan pemrosesan bahasa alami (Alzubaidi et al., 2021). Perbedaan antara *machine learning* dan *deep learning* dapat dilihat pada Gambar 3.6.




**Gambar 3.6.** Perbedaan DL dan ML (Alzubaidi et al., 2021)

## 3.5. *Convolutional Neural Network* (CNN)

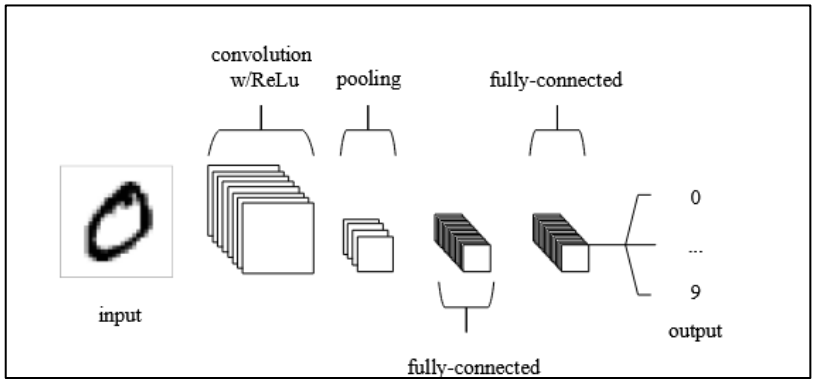
*Convolutional Neural Network* (CNN) merupakan salah satu arsitektur *deep learning* yang mampu mengekstraksi fitur secara otomatis dari data sehingga dapat beradaptasi terhadap variasi karakteristik data yang sulit direpresentasikan menggunakan fitur buatan tangan (Wang et al., 2020). CNN merupakan arsitektur jaringan saraf yang dirancang untuk menangani variasi pola dua dimensi seperti data citra. CNN bekerja langsung pada data piksel sehingga proses ekstraksi fitur tidak lagi sepenuhnya bergantung pada fitur yang dirancang secara manual atau *handcrafted features* (LeCun et al., 1998).

Arsitektur CNN terinspirasi oleh sistem visual pada otak manusia dan hewan, khususnya visual cortex. Arsitektur ini mampu mempelajari korelasi lokal pada citra melalui penerapan koneksi lokal dan *parameter sharing*. Berbeda dengan jaringan saraf *fully connected*, CNN menggunakan bobot bersama sehingga jumlah parameter yang perlu dipelajari menjadi lebih sedikit. Pendekatan tersebut menyederhanakan proses pelatihan dan mempercepat komputasi. Selain itu, penggunaan *shared weights* juga meningkatkan kemampuan generalisasi model serta mengurangi risiko *overfitting* (Alzubaidi et al., 2021). Dibandingkan jaringan

saraf tradisional, CNN memiliki keunggulan berupa ketahanan terhadap pergeseran dan distorsi citra, kebutuhan memori yang lebih rendah, serta proses pelatihan yang lebih efisien sehingga banyak digunakan dalam berbagai tugas pengenalan pola dan klasifikasi (Hijazi et al., 2015). Hal ini menjadikan CNN sangat sesuai untuk tugas *computer vision*, termasuk pengenalan wajah dan deteksi *deepfake*.

### 3.6. Arsitektur *Convolutional Neural Network* (CNN)

CNN dikembangkan untuk memproses data citra, sehingga arsitekturnya dirancang agar sesuai dengan karakteristik gambar sebagai masukan. Salah satu perbedaan utama CNN dibandingkan ANN tradisional adalah representasi data pada setiap lapisannya yang berbentuk volume tiga dimensi, yaitu tinggi (*height*), lebar (*width*), dan kedalaman (*depth*). Dimensi kedalaman tidak menunjukkan jumlah lapisan pada jaringan, melainkan dimensi ketiga dari *activation volume*. Selain itu, berbeda dengan ANN yang menerapkan koneksi penuh (*fully connected*), setiap neuron pada suatu lapisan CNN hanya terhubung dengan sebagian kecil wilayah pada lapisan sebelumnya atau *local connectivity*. Pendekatan ini memungkinkan jumlah parameter yang digunakan menjadi lebih sedikit sehingga kompleksitas model dapat dikurangi. CNN tersusun atas tiga jenis lapisan utama, yaitu *convolutional layer*, *pooling layer*, dan *fully connected layer*. Penyusunan ketiga lapisan tersebut secara berurutan membentuk suatu arsitektur CNN (O'shea & Nash, 2015). Sebuah contoh arsitektur CNN yang disederhanakan ditunjukkan pada Gambar 3.7.




**Gambar 3.7.** Contoh Arsitektur CNN Sederhana (O'shea & Nash, 2015)

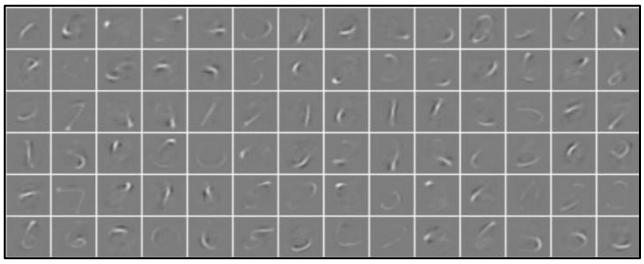
Fungsi dasar dari contoh CNN pada Gambar 3.7 dapat dibagi ke dalam empat bagian utama sebagai berikut.

1. Sebagaimana pada bentuk ANN lainnya, lapisan masukan (*input layer*) menyimpan nilai piksel dari citra yang digunakan sebagai data masukan.
2. Lapisan konvolusi (*convolutional layer*) menentukan keluaran neuron-neuron yang terhubung dengan wilayah lokal tertentu dari data masukan melalui perhitungan hasil kali skalar antara bobot neuron dan bagian *input* yang bersesuaian pada volume masukan. Selanjutnya, fungsi aktivasi *Rectified Linear Unit* (ReLU) diterapkan secara elemen-per-elemen (*element-wise*) terhadap keluaran yang dihasilkan oleh lapisan sebelumnya.
3. Lapisan *pooling* berfungsi untuk melakukan proses *downsampling* pada dimensi spasial data masukan sehingga jumlah parameter pada volume aktivasi dapat dikurangi.
4. Lapisan *fully connected* menjalankan fungsi yang serupa dengan ANN konvensional, yaitu menghasilkan skor kelas berdasarkan aktivasi yang diperoleh, yang kemudian digunakan untuk proses klasifikasi.

Melalui rangkaian transformasi tersebut, CNN mampu memproses data masukan secara bertahap menggunakan operasi konvolusi dan *downsampling* untuk menghasilkan skor kelas yang dapat digunakan dalam tugas klasifikasi (O'shea & Nash, 2015).

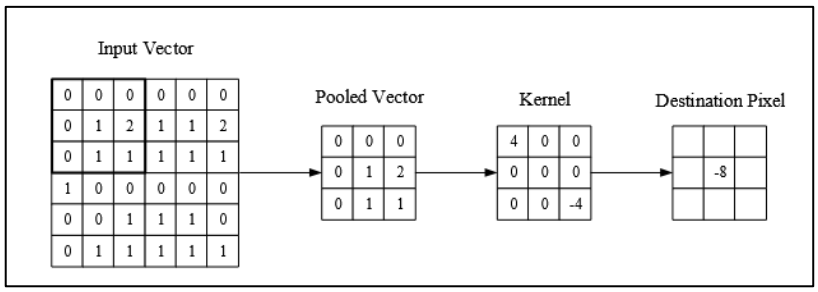
### 3.6.1. *Convolutional layer*

Sesuai dengan namanya, *convolutional layer* memegang peranan yang sangat penting dalam cara kerja CNN. Parameter pada lapisan ini berfokus pada penggunaan kernel yang dapat dipelajari. Kernel-kernel tersebut umumnya berukuran kecil pada dimensi spasial, namun mencakup seluruh kedalaman dari data masukan. Ketika data masukan diproses oleh *convolutional layer*, setiap kernel akan dikonvolusikan melintasi dimensi spasial dari *input* untuk menghasilkan peta aktivasi dua dimensi. Peta-peta aktivasi tersebut dapat divisualisasikan, sebagaimana ditunjukkan pada Gambar 3.8.



**Gambar 3.8.** Contoh Peta Aktivasi (O'shea & Nash, 2015)

Selama proses konvolusi, kernel digeser secara sistematis di sepanjang citra masukan dan pada setiap pergeseran dihitung hasil kali skalar antara nilai kernel dan bagian citra yang bersesuaian. Melalui proses tersebut, jaringan saraf mempelajari kernel yang akan memberikan respons ketika mendeteksi fitur tertentu pada posisi spasial tertentu dalam citra masukan. Hasil dari proses ini disebut aktivasi, sedangkan setiap kernel menghasilkan sebuah peta aktivasi. Kumpulan peta aktivasi tersebut kemudian disusun pada dimensi kedalaman (*depth*) untuk membentuk volume keluaran dari *convolutional layer*. Contoh representasi *convolutional layer* dapat dilihat pada Gambar 3.9.




**Gambar 3.9.** Contoh *Convolutional Layer* (O'shea & Nash, 2015)

Pada lapisan konvolusi, setiap kernel menghasilkan sebuah peta aktivasi. Seluruh peta aktivasi tersebut kemudian disusun pada dimensi kedalaman sehingga membentuk volume keluaran dari *convolutional layer*. Penggunaan *convolutional layer* dikembangkan untuk mengatasi keterbatasan ANN konvensional yang menggunakan koneksi penuh, yang menyebabkan jumlah parameter menjadi sangat besar ketika memproses data citra. Oleh karena itu, setiap neuron pada *convolutional layer* hanya terhubung dengan sebagian kecil wilayah pada volume masukan yang disebut *receptive field*. Besarnya konektivitas pada dimensi kedalaman umumnya sama dengan kedalaman volume masukan. Dimensi keluaran yang dihasilkan oleh operasi konvolusi dipengaruhi oleh ukuran *input*, ukuran *receptive field*, *padding*, dan *stride*. Hubungan tersebut dapat dihitung menggunakan persamaan berikut:

$$O = \frac{(V - R) + 2Z}{S} + 1 \tag{3.1}$$

di mana $V$ adalah ukuran *input*, $R$ adalah ukuran *receptive field*, $Z$ adalah jumlah *zero padding*, dan $S$ adalah nilai *stride*. Apabila hasil perhitungan bukan berupa bilangan bulat, maka nilai *stride* yang digunakan tidak sesuai karena *receptive field*

tidak dapat ditempatkan secara tepat pada seluruh area masukan (O'shea & Nash, 2015).

### 3.6.2. Fungsi Aktivasi

Setelah operasi konvolusi, *feature map* yang dihasilkan diteruskan ke fungsi aktivasi non-linear untuk meningkatkan kemampuan jaringan dalam mempelajari representasi yang lebih kompleks. Salah satu fungsi aktivasi yang paling umum digunakan pada CNN adalah *Rectified Linear Unit* (ReLU), yang didefinisikan sebagai:

$$f(x) = max(0, x) \tag{3.2}$$

di mana $x$ merupakan nilai masukan fungsi aktivasi dan $f(x)$ adalah nilai keluaran setelah penerapan fungsi ReLU. Fungsi ini menghasilkan nilai keluaran yang sama dengan nilai masukan apabila $x$ bernilai positif, sedangkan apabila $x$ bernilai nol atau negatif, maka nilai keluarannya adalah nol (I. Goodfellow et al., 2016; Nair & Hinton, 2010).

### 3.6.3. *Pooling Layer*

*Pooling layer* merupakan lapisan yang berfungsi untuk mengurangi dimensi spasial dari *feature map* yang dihasilkan oleh *convolutional layer*. Proses ini dilakukan secara bertahap untuk mengurangi jumlah parameter dan beban komputasi pada jaringan, sehingga model menjadi lebih efisien. Selain itu, *pooling* juga membantu meningkatkan invariansi terhadap perubahan kecil pada posisi objek dalam citra, sehingga model tetap mampu mengenali fitur yang sama meskipun terjadi pergeseran lokasi pada *input* (I. Goodfellow et al., 2016).

Salah satu metode *pooling* yang paling umum digunakan pada CNN adalah *max pooling*. Metode ini bekerja dengan membagi *feature map* menjadi beberapa wilayah lokal (*pooling region*), kemudian memilih nilai maksimum dari setiap wilayah tersebut sebagai representasi keluaran. Operasi ini diterapkan secara independen pada setiap *feature map* sehingga jumlah *feature map* tetap sama, sedangkan dimensi spasialnya menjadi lebih kecil. Pada umumnya, *max pooling* menggunakan *kernel* berukuran 2 × 2 dengan nilai *stride* sebesar 2, sehingga ukuran *feature map* setelah proses *pooling* menjadi sekitar seperempat dari ukuran sebelumnya (I. Goodfellow et al., 2016; O'shea & Nash, 2015).

### 3.6.4. *Fully Connected Layer*

Lapisan *fully connected* merupakan lapisan yang menghubungkan setiap neuron pada lapisan sebelumnya dengan seluruh neuron pada lapisan berikutnya. Pada CNN, lapisan ini menerima fitur hasil ekstraksi dari *convolutional layer* dan *pooling layer* yang telah diubah menjadi vektor satu dimensi melalui proses *flatten*. Selanjutnya, lapisan *fully connected* menghasilkan skor kelas yang digunakan sebagai dasar dalam proses klasifikasi (O'shea & Nash, 2015). Secara matematis, operasi pada *fully connected layer* merupakan transformasi linear yang dapat dinyatakan dalam bentuk vektor sebagai:

$$\alpha = Wx + b \tag{3.3}$$

di mana $\alpha$ adalah vektor hasil transformasi linear sebelum diterapkan fungsi aktivasi, $W$ adalah matriks bobot yang merepresentasikan hubungan antar neuron, $x$ adalah vektor masukan yang berasal dari lapisan sebelumnya, dan $b$ adalah vektor bias. Persamaan tersebut merupakan bentuk vektor dari proses penjumlahan berbobot (*weighted sum*) antara *input* dan bobot yang dijelaskan pada model jaringan saraf tiruan, sebelum keluaran diteruskan ke fungsi aktivasi (Bishop, 2006; I. Goodfellow et al., 2016).

### 3.6.5. *Output Layer* dan Klasifikasi

Pada tugas klasifikasi multikelas, lapisan *output* pada jaringan saraf umumnya menggunakan fungsi *Softmax* untuk mengubah nilai keluaran dari *layer* sebelumnya menjadi distribusi probabilitas pada setiap kelas. Nilai probabilitas yang dihasilkan berada pada rentang 0 hingga 1, dengan jumlah probabilitas seluruh kelas sama dengan satu (K. H. Mahmud et al., 2019). Pada CNN, fungsi *Softmax* umumnya diterapkan setelah *fully connected layer* sebagai tahap akhir proses klasifikasi (Krizhevsky et al., 2012).

$$Softmax(\alpha_i) = \frac{e^{\alpha_i}}{\sum_{j=1}^{K} e^{\alpha_j}} \tag{3.4}$$

di mana $\alpha_i$ adalah nilai input ke-i pada lapisan *output* sebelum fungsi aktivasi (*logit*), $e$ adalah bilangan Euler ($\approx 2.71828$), $K$ adalah jumlah kelas, $\sum_{j=1}^{K} e^{\alpha_j}$ adalah jumlah eksponensial seluruh nilai *input* pada lapisan *output*, dan $Softmax(\alpha_i)$ adalah probabilitas prediksi untuk kelas ke-i. Fungsi *Softmax* mengubah nilai logit menjadi

distribusi probabilitas sehingga kelas dengan probabilitas tertinggi dipilih sebagai hasil klasifikasi (Bishop, 2006).

### 3.7. EfficientNet-B0

Arsitektur CNN konvensional umumnya meningkatkan performa dengan memperbesar ukuran model melalui penambahan kedalaman atau lebar jaringan. Meskipun pendekatan tersebut dapat meningkatkan akurasi, peningkatan ukuran model juga menyebabkan bertambahnya kebutuhan komputasi, meningkatkan risiko *overfitting*, serta belum tentu memberikan peningkatan performa yang sebanding akibat munculnya kesulitan optimisasi pada jaringan yang semakin dalam (He et al., 2016; Szegedy et al., 2015).

Berdasarkan permasalahan tersebut, Tan dan Le (2019) mengembangkan EfficientNet sebagai keluarga arsitektur *Convolutional Neural Network* (CNN) yang dirancang untuk meningkatkan akurasi sekaligus efisiensi komputasi melalui metode penskalaan model yang lebih sistematis. Berbeda dengan pendekatan *scaling* konvensional yang umumnya hanya meningkatkan salah satu dimensi jaringan, yaitu kedalaman (*depth*), lebar (*width*), atau resolusi citra masukan secara terpisah, EfficientNet memperkenalkan metode *compound scaling* yang melakukan penskalaan ketiga dimensi tersebut secara bersamaan menggunakan rasio yang tetap. Pendekatan ini bertujuan menghasilkan keseimbangan yang lebih baik antara akurasi dan efisiensi model.

EfficientNet-B0 merupakan model dasar (*baseline*) dalam keluarga EfficientNet yang diperoleh melalui proses *neural architecture search* (NAS). Selanjutnya, model ini digunakan sebagai dasar untuk membangun varian EfficientNet-B1 hingga EfficientNet-B7 dengan menerapkan metode *compound scaling* menggunakan koefisien yang berbeda. Berdasarkan hasil evaluasi pada dataset ImageNet, keluarga EfficientNet mampu mencapai akurasi yang tinggi dengan jumlah parameter dan kebutuhan komputasi yang lebih rendah dibandingkan beberapa arsitektur CNN populer, seperti ResNet, DenseNet, dan Inception (Tan & Le, 2019).

#### 3.7.1. Arsitektur EfficientNet-B0

Arsitektur EfficientNet-B0 tersusun atas beberapa *stage* yang menggunakan *Mobile Inverted Bottleneck Convolution* (MBConv) sebagai blok penyusun utama.

Setiap tahap memiliki konfigurasi yang berbeda, meliputi jenis operator, ukuran kernel, faktor ekspansi (*expansion ratio*), jumlah kanal keluaran (*output channels*), jumlah pengulangan blok (*repeats*), serta resolusi *feature map*. Variasi konfigurasi tersebut memungkinkan jaringan mengekstraksi fitur secara bertahap, mulai dari fitur tingkat rendah hingga fitur tingkat tinggi, dengan tetap mempertahankan efisiensi komputasi. Selain menggunakan blok MBConv, EfficientNet-B0 juga mengintegrasikan *Squeeze-and-Excitation (SE) block* pada setiap blok MBConv untuk meningkatkan representasi fitur melalui pemodelan hubungan antar-channel. Kombinasi MBConv dan *SE block* menghasilkan arsitektur yang mampu mengekstraksi fitur secara efektif dengan jumlah parameter yang relatif sedikit. Konfigurasi lengkap setiap tahap pada arsitektur EfficientNet-B0 disajikan pada Tabel 3.1 (Tan & Le, 2019).

**Tabel 3.1.** EfficientNet-B0 *Baseline Network*

| *Stage* $i$ | *Operator* $\hat{F}_i$ | *Resolution* $\hat{H}_i \times \hat{W}_i$ | *#Channels* $\hat{C}_i$ | *#Layers* $\hat{L}_i$ |
|---|---|---|---|---|
| 1 | Conv3x3 | 224 × 224 | 32 | 1 |
| 2 | MBConv1, k3x3 | 112 × 112 | 16 | 1 |
| 3 | MBConv6, k3x3 | 112 × 112 | 24 | 2 |
| 4 | MBConv6, k5x5 | 56 × 56 | 40 | 2 |
| 5 | MBConv6, k3x3 | 28 × 28 | 80 | 3 |
| 6 | MBConv6, k5x5 | 14 × 14 | 112 | 3 |
| 7 | MBConv6, k5x5 | 14 × 14 | 192 | 4 |
| 8 | MBConv6, k3x3 | 7 × 7 | 320 | 1 |
| 9 | Conv1x1 & Pooling & FC | 7 × 7 | 1280 | 1 |

Berdasarkan Tabel 3.1, EfficientNet-B0 terdiri atas beberapa blok MBConv yang disusun secara bertingkat dengan konfigurasi yang berbeda pada setiap tahap jaringan. Perbedaan konfigurasi tersebut mencakup ukuran kernel, faktor ekspansi, jumlah kanal keluaran, jumlah pengulangan blok, dan resolusi *feature map*, yang secara bersama-sama mendukung proses ekstraksi fitur secara efisien pada berbagai tingkat representasi (Tan & Le, 2019).

### 3.7.2. *Compound Scaling Method*

Konsep utama EfficientNet terletak pada *compound scaling method*, yaitu strategi penskalaan jaringan yang mengatur kedalaman, lebar, dan resolusi *input* secara proporsional menggunakan satu koefisien skala. Metode ini dirumuskan sebagai:

$$depth: d = \alpha^{\emptyset} \tag{3.5}$$

$$width: w = \beta^{\emptyset} \tag{3.6}$$

$$resolution: r = \gamma^{\emptyset} \tag{3.7}$$

dengan kendala:

$$\alpha \cdot \beta^2 \cdot \gamma^2 \approx 2 \tag{3.8}$$

di mana $\alpha, \beta, \gamma$ adalah konstanta penskalaan untuk kedalaman, lebar, resolusi, dan $\emptyset$ adalah koefisien skala global yang mengontrol ukuran model. Formulasi tersebut dirancang untuk menyeimbangkan peningkatan kedalaman, lebar, dan resolusi jaringan sehingga model dapat mencapai akurasi yang lebih baik dengan efisiensi komputasi yang tetap terjaga (Tan & Le, 2019).

### 3.7.3. *Mobile Inverted Bottleneck Convolution*

EfficientNet-B0 menggunakan blok utama yang disebut *Mobile Inverted Bottleneck Convolution* (MBConv), yaitu blok konvolusi yang diadaptasi dari arsitektur MobileNetV2 (Tan & Le, 2019). MBConv dirancang untuk meningkatkan efisiensi komputasi melalui kombinasi *depthwise separable convolution*, *inverted residual connection*, dan *linear bottleneck*. Struktur tersebut memungkinkan jaringan menghasilkan representasi fitur yang baik dengan jumlah parameter dan kebutuhan komputasi yang lebih rendah dibandingkan blok konvolusi konvensional (Sandler et al., 2018).

Salah satu komponen utama MBConv adalah *depthwise separable convolution*. Teknik ini memisahkan operasi konvolusi standar menjadi dua tahap, yaitu *depthwise convolution* dan *pointwise convolution*. Pada tahap *depthwise convolution*, satu filter diterapkan pada setiap kanal masukan (*input channel*) secara

terpisah sehingga proses ekstraksi fitur dilakukan secara independen pada masing-masing kanal. Selanjutnya, hasil *depthwise convolution* diproses menggunakan *pointwise convolution*, yaitu konvolusi berukuran 1×1 yang mengombinasikan informasi dari seluruh kanal untuk membentuk representasi fitur baru. *Depthwise convolution* dengan satu filter per saluran masukan (*input depth*) dapat ditulis sebagai:

$$\hat{G}_{k,l,m} = \sum_{i,j} \hat{K}_{i,j,m} F_{k+i-1,l+j-1,m} \tag{3.9}$$

dengan $\hat{G}$ merupakan *feature map* keluaran hasil *depthwise convolution*, $F$ adalah *feature map* masukan, $\hat{K}$ merupakan *kernel depthwise* pada kanal ke-m, sedangkan $i$ dan $j$ menyatakan indeks *kernel* konvolusi. Persamaan tersebut menunjukkan bahwa setiap kanal masukan diproses menggunakan satu *kernel* secara independen tanpa menggabungkan informasi antar-kanal (Howard et al., 2017).

Howard et al. (2017) juga menjelaskan bahwa penggunaan *depthwise separable convolution* mampu menurunkan kompleksitas komputasi dibandingkan konvolusi standar. Kompleksitas komputasi konvolusi standar dinyatakan sebagai:

$$D_K^2 \cdot M \cdot N \cdot D_F^2 \tag{3.10}$$

sedangkan kompleksitas komputasi *depthwise separable convolution* dinyatakan sebagai:

$$D_K^2 \cdot M \cdot D_F^2 + M \cdot N \cdot D_F^2 \tag{3.11}$$

dengan $D_K$ menyatakan ukuran *kernel* konvolusi, $M$ jumlah kanal masukan, $N$ jumlah kanal keluaran, dan $D_F$ ukuran spasial *feature map*. Pemisahan operasi konvolusi menjadi tahap *depthwise* dan *pointwise* tersebut mampu mengurangi kebutuhan komputasi secara signifikan dibandingkan konvolusi standar sehingga menghasilkan model yang lebih efisien.

### 3.7.4. *Squeeze-and-Excitation Block*

EfficientNet-B0 mengintegrasikan *Squeeze-and-Excitation (SE) block* sebagai bagian dari blok MBConv untuk meningkatkan kemampuan representasi fitur melalui pemodelan hubungan antar-channel. *SE block* secara adaptif melakukan *feature recalibration* dengan memanfaatkan informasi global dari setiap *channel* sehingga jaringan dapat memberikan bobot yang lebih besar pada fitur yang lebih informatif (Hu et al., 2019; Tan & Le, 2019). Pada tahap *squeeze*, setiap

*channel* diringkas menggunakan operasi *global average pooling* yang dirumuskan sebagai:

$$z_c = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} u_c(i,j) \tag{3.12}$$

Selanjutnya, tahap *excitation* mempelajari hubungan antar-channel melalui dua lapisan *fully connected* yang dipisahkan oleh fungsi aktivasi ReLU dan diakhiri fungsi sigmoid sehingga menghasilkan bobot perhatian (*attention weights*) untuk setiap *channel*. Proses tersebut dirumuskan sebagai

$$s = \sigma\big(W_2 \delta(W_1 z)\big) \tag{3.13}$$

di mana $z$ merupakan vektor hasil *global average pooling*, $W_1$ dan $W_2$ adalah matriks bobot pada lapisan *fully connected*, $\delta(\cdot)$ adalah fungsi aktivasi ReLU, dan $\sigma(\cdot)$ adalah fungsi sigmoid. Bobot yang dihasilkan kemudian digunakan untuk melakukan *recalibration* terhadap setiap *channel* sehingga fitur yang lebih informatif memperoleh perhatian lebih besar dibandingkan fitur yang kurang relevan (Hu et al., 2019).

### 3.8. *Confusion Matrix*

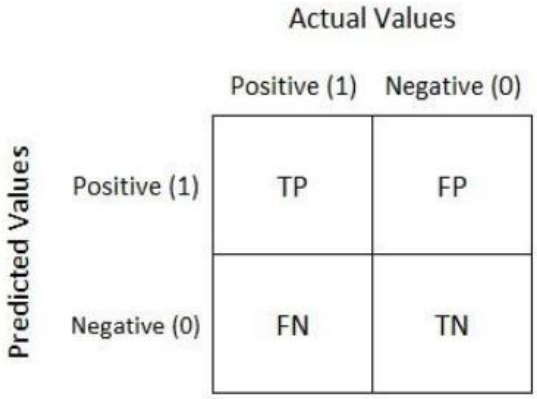
*Confusion matrix* merupakan tabel evaluasi yang digunakan untuk membandingkan hasil prediksi model dengan label sebenarnya pada suatu tugas klasifikasi. Pada klasifikasi biner, *confusion matrix* terdiri atas empat komponen utama, yaitu *true positive* (TP), *true negative* (TN), *false positive* (FP), dan *false negative* (FN), yang menunjukkan jumlah prediksi benar maupun salah yang dihasilkan model. Pada klasifikasi multikelas, *confusion matrix* diperluas sehingga setiap baris merepresentasikan kelas sebenarnya, sedangkan setiap kolom menunjukkan kelas hasil prediksi. Informasi pada *confusion matrix* menjadi dasar dalam perhitungan berbagai metrik evaluasi, seperti *accuracy*, *precision*, *recall*, *specificity*, dan *F1-score*, sehingga memungkinkan evaluasi kinerja model secara lebih komprehensif pada tugas klasifikasi biner maupun multikelas (Sokolova & Lapalme, 2009).

Selain menjadi dasar dalam perhitungan berbagai metrik evaluasi, informasi yang diperoleh dari *confusion matrix* memungkinkan analisis terhadap pola kesalahan klasifikasi yang dilakukan oleh model. Analisis tersebut menjadi penting terutama pada kondisi *dataset* yang tidak seimbang, karena evaluasi yang hanya

didasarkan pada nilai *accuracy* belum tentu mencerminkan kemampuan model dalam membedakan setiap kelas secara tepat. Oleh karena itu, metrik evaluasi yang diturunkan dari *confusion matrix* memberikan gambaran yang lebih representatif mengenai performa model dibandingkan hanya mengandalkan nilai *accuracy* (Fawcett, 2006).

### 3.8.1. Struktur *Confusion Matrix*

Pada klasifikasi biner, *confusion matrix* direpresentasikan dalam bentuk matriks berukuran 2×2 yang terdiri atas empat komponen utama, yaitu *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN). Keempat komponen tersebut menunjukkan jumlah prediksi yang benar maupun salah yang dihasilkan oleh model klasifikasi dan membentuk struktur dasar *confusion matrix* sebagai representasi hasil klasifikasi (Sokolova & Lapalme, 2009).




**Gambar 3.10.** *Confusion Matrix* (Yoga Wibowo et al., 2023)

Sel pada diagonal utama *confusion matrix* menunjukkan prediksi yang benar, yaitu *True Positive* (TP) dan *True Negative* (TN). *True Positive* menyatakan jumlah data kelas positif yang berhasil diprediksi sebagai positif, sedangkan *True Negative* menyatakan jumlah data kelas negatif yang berhasil diprediksi sebagai negatif. Sebaliknya, sel di luar diagonal utama menunjukkan kesalahan klasifikasi, yaitu *False Positive* (FP), ketika data kelas negatif diprediksi sebagai positif, dan *False Negative* (FN), ketika data kelas positif diprediksi sebagai negatif (Powers, 2011; Sokolova & Lapalme, 2009).

Konsep *confusion matrix* tidak hanya diterapkan pada klasifikasi biner, tetapi juga dapat diperluas pada klasifikasi multikelas dengan mengevaluasi setiap kelas secara individual. Pada klasifikasi multikelas, nilai *True Positive*, *True Negative*,

*False Positive*, dan *False Negative* dihitung untuk masing-masing kelas, sehingga berbagai metrik evaluasi tetap dapat diperoleh berdasarkan komponen *confusion matrix*. Pendekatan ini memungkinkan analisis performa model dilakukan secara lebih rinci pada setiap kelas yang diklasifikasikan, sehingga sesuai digunakan untuk mengevaluasi model pada permasalahan klasifikasi multikelas (Sokolova & Lapalme, 2009).

### 3.8.2. *Accuracy*

*Accuracy* merupakan metrik evaluasi yang mengukur proporsi prediksi yang benar terhadap seluruh data uji. Secara matematis, *accuracy* dirumuskan sebagai:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.14}$$

Rumus tersebut menunjukkan tingkat ketepatan keseluruhan model dalam mengklasifikasikan data berdasarkan jumlah prediksi yang benar dibandingkan dengan seluruh data yang dievaluasi (Powers, 2011; Sokolova & Lapalme, 2009).

### 3.8.3. *Precision*

*Precision* mengukur tingkat ketepatan model dalam memprediksi kelas positif, yaitu proporsi prediksi positif yang benar terhadap seluruh prediksi positif. Secara matematis, *precision* dirumuskan sebagai:

$$Precision = \frac{TP}{TP + FP} \tag{3.15}$$

Metrik *precision* menunjukkan kemampuan model dalam meminimalkan jumlah *false positive*. Semakin tinggi nilai *precision*, maka semakin sedikit prediksi positif yang sebenarnya merupakan kelas negatif (Powers, 2011). Metrik *precision* sangat penting pada kasus di mana kesalahan *false positive* harus diminimalkan, seperti pada sistem keamanan dan deteksi *deepfake* di mana kesalahan mengklasifikasikan wajah asli sebagai *deepfake* dapat menimbulkan konsekuensi serius.

### 3.8.4. *Recall (Sensitivity)*

*Recall* yang juga dikenal sebagai *sensitivity* atau *true positive rate* (TPR) merupakan metrik evaluasi yang mengukur kemampuan model dalam mengidentifikasi seluruh data yang termasuk ke dalam kelas positif. Secara matematis, *recall* dirumuskan sebagai:

$$Recall = \frac{TP}{TP + FN} \tag{3.16}$$

Semakin tinggi nilai *recall*, semakin besar proporsi data positif yang berhasil diidentifikasi oleh model, sehingga menunjukkan kemampuan model dalam meminimalkan kesalahan *false negative* (Fawcett, 2006; Sokolova & Lapalme, 2009). Dalam konteks deteksi *deepfake*, *recall* penting untuk memastikan bahwa sebagian besar wajah *deepfake* dapat teridentifikasi oleh sistem.

### 3.8.5. *F1-Score*

*F1-score* merupakan salah satu metrik evaluasi yang berasal dari keluarga *F-score* dan digunakan untuk mengukur kinerja model klasifikasi dengan menggabungkan nilai *precision* dan *recall* ke dalam satu ukuran. *F1-score* merupakan bentuk khusus dari *F-score* dengan parameter $\beta = 1$, sehingga *precision* dan *recall* memiliki bobot yang sama dalam proses evaluasi. Secara matematis, *F1-score* dirumuskan sebagai:

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{3.17}$$

Dengan menggabungkan *precision* dan *recall*, *F1-score* memberikan ukuran evaluasi yang mempertimbangkan kedua metrik tersebut secara bersamaan, sehingga dapat digunakan untuk menilai kinerja model klasifikasi secara lebih komprehensif (Altuncu et al., 2024).

## 3.9. Kurva ROC dan Skor AUC

Kurva *Receiver Operating Characteristic* (ROC) merupakan alat evaluasi yang digunakan untuk menilai kinerja model klasifikasi dengan memvisualisasikan hubungan antara *True Positive Rate* (TPR) dan *False Positive Rate* (FPR) pada berbagai nilai ambang batas (*threshold*). ROC pertama kali diperkenalkan dalam teori deteksi sinyal dan kemudian diadopsi secara luas dalam bidang *machine learning* untuk mengevaluasi performa model klasifikasi (Fawcett, 2006). Pada kurva ROC, sumbu vertikal merepresentasikan *True Positive Rate* (TPR) atau *recall*, sedangkan sumbu horizontal menunjukkan *False Positive Rate* (FPR). Nilai TPR dan FPR dirumuskan sebagai berikut:

$$TPR = \frac{TP}{TP + FN} \tag{3.18}$$

$$FPR = \frac{FP}{FP + TN} \tag{3.19}$$

dengan TP adalah *True Positive*, FN adalah *False Negative*, FP adalah *False Positive*, dan TN adalah *True Negative*. Nilai TPR menunjukkan kemampuan model dalam mengidentifikasi sampel positif secara benar (*recall*), sedangkan FPR menunjukkan proporsi sampel negatif yang salah diklasifikasikan sebagai positif (Fawcett, 2006; Sokolova & Lapalme, 2009).

### 3.9.1. *Area Under the Curve* (AUC)

*Area Under the Curve* (AUC) merupakan ukuran numerik yang merepresentasikan luas area di bawah kurva *Receiver Operating Characteristic* (ROC). AUC digunakan sebagai ukuran tunggal untuk mengevaluasi kinerja model klasifikasi berdasarkan kurva ROC. Nilai AUC berada pada rentang 0 hingga 1, di mana nilai yang lebih tinggi menunjukkan kemampuan model yang lebih baik dalam membedakan antara kelas positif dan kelas negatif. Selain itu, nilai AUC sebesar 0.5 menunjukkan performa yang setara dengan klasifikasi acak (*random guessing*), sedangkan nilai yang mendekati 1 menunjukkan kemampuan diskriminasi yang semakin baik. Secara matematis, AUC dapat diinterpretasikan sebagai probabilitas bahwa model memberikan skor yang lebih tinggi pada suatu sampel positif yang dipilih secara acak dibandingkan dengan sampel negatif yang juga dipilih secara acak (Fawcett, 2006). Dalam praktiknya, AUC dihitung sebagai luas area di bawah kurva ROC yang secara matematis dapat dinyatakan sebagai berikut:

$$AUC = \int_0^1 TPR(FPR)d(FPR) \tag{3.20}$$

Rumus ini menunjukkan bahwa AUC merupakan agregasi performa model pada seluruh kemungkinan *threshold* (Nurfansepta et al., 2025).

### 3.9.2. ROC dan AUC pada Klasifikasi Multikelas

Pada kasus klasifikasi multikelas, *Receiver Operating Characteristic* (ROC) dan *Area Under the Curve* (AUC) yang pada dasarnya dikembangkan untuk klasifikasi biner tidak dapat diterapkan secara langsung karena melibatkan lebih dari dua kelas. Oleh karena itu, diperlukan perluasan metode AUC sehingga dapat digunakan sebagai ukuran evaluasi pada klasifikasi multikelas. Secara umum,

terdapat dua pendekatan utama yang digunakan, yaitu *One-versus-One* (OvO) dan *One-versus-Rest* (OvR).

Pada pendekatan *One-versus-One* (OvO), nilai AUC diperoleh dengan menghitung AUC untuk setiap pasangan kelas (*pairwise comparison*), kemudian seluruh nilai AUC pasangan tersebut dirata-ratakan sehingga diperoleh satu nilai AUC multikelas. Pendekatan ini mengevaluasi kemampuan model dalam membedakan setiap pasangan kelas secara terpisah. Sementara itu, pendekatan *One-versus-Rest* (OvR) menganggap bahwa suatu permasalahan klasifikasi yang memiliki c kelas dapat direpresentasikan sebagai c buah klasifikasi biner yang berjalan secara paralel. Pada setiap proses klasifikasi, satu kelas diperlakukan sebagai kelas positif (target), sedangkan seluruh kelas lainnya digabungkan menjadi kelas negatif (non-target). Nilai AUC kemudian dihitung untuk masing-masing kelas terhadap seluruh kelas lainnya dan selanjutnya dirata-ratakan sehingga diperoleh nilai AUC multikelas yang dirumuskan sebagai berikut:

$$AUC_{OvR} = \frac{1}{c}\sum_{i=0}^{c-1} AUC\left(C_i, C_i^C\right) \tag{3.21}$$

dengan $c$ adalah jumlah kelas, $C_i$ merupakan himpunan data pada kelas ke-i, sedangkan $C_i^C$ adalah himpunan komplemen yang terdiri atas seluruh data selain kelas ke-i. Nilai $AUC\left(C_i, C_i^C\right)$ menunjukkan nilai AUC yang diperoleh ketika kelas ke-i dianggap sebagai kelas positif dan seluruh kelas lainnya sebagai kelas negatif. Pendekatan OvR memiliki kompleksitas komputasi yang lebih rendah dibandingkan OvO. Pada pendekatan OvO diperlukan perhitungan AUC untuk seluruh pasangan kelas sehingga kompleksitasnya meningkat secara kuadratik terhadap jumlah kelas, sedangkan pada pendekatan OvR hanya diperlukan perhitungan sebanyak jumlah kelas sehingga kompleksitasnya meningkat secara linear. Oleh karena itu, pendekatan OvR lebih efisien untuk diterapkan pada klasifikasi multikelas dengan jumlah kelas yang relatif banyak (Gimeno et al., 2021).

### 3.9.3. Interpretasi Nilai AUC

Nilai *Area Under the Curve* (AUC) digunakan untuk mengukur kemampuan model klasifikasi dalam membedakan kelas positif dan kelas negatif secara keseluruhan. Nilai AUC berada pada rentang 0 hingga 1, di mana nilai yang

semakin besar menunjukkan kemampuan diskriminasi model yang semakin baik. Suatu model dikatakan memiliki kemampuan diskriminasi yang bermakna apabila nilai AUC lebih besar dari 0.5. Secara umum, nilai AUC sebesar 0.8 atau lebih dianggap telah memiliki performa yang dapat diterima (*acceptable*) dalam membedakan kedua kelas (Nahm, 2022).

**Tabel 3.2.** Interpretasi Nilai AUC

| Nilai AUC | Interpretasi Kinerja Model |
|---|---|
| $0.9 \leq AUC$ | Sangat baik (*Excellent*) |
| $0.8 \leq AUC < 0.9$ | Baik (*Good*) |
| $0.7 \leq AUC < 0.8$ | Cukup (*Fair*) |
| $0.6 \leq AUC < 0.7$ | Buruk (*Poor*) |
| $0.5 \leq AUC < 0.6$ | Sangat buruk (*Fail*) |

Berdasarkan klasifikasi tersebut, model dengan nilai AUC yang lebih tinggi memiliki kemampuan yang lebih baik dalam membedakan kelas positif dan kelas negatif. Sebaliknya, apabila nilai AUC mendekati 0.5 maka kemampuan model dalam melakukan klasifikasi tidak lebih baik dibandingkan klasifikasi secara acak (*random guessing*) (Nahm, 2022).

### 3.10. Visualisasi Ruang Fitur dengan t-SNE

Visualisasi ruang fitur (*feature space visualization*) digunakan untuk membantu memahami representasi fitur yang dihasilkan oleh model pembelajaran mendalam. Pada *Convolutional Neural Network* (CNN), citra masukan diproses melalui serangkaian lapisan konvolusi sehingga jaringan secara otomatis mempelajari representasi fitur yang semakin kaya pada setiap lapisan. CNN mampu membangun *feature extractor* secara otomatis sehingga representasi yang dihasilkan dapat digunakan untuk proses klasifikasi (LeCun et al., 1998).

Karena representasi fitur tersebut umumnya berada pada ruang berdimensi tinggi, diperlukan metode reduksi dimensi agar hubungan antar data dapat divisualisasikan dalam ruang dua atau tiga dimensi. Salah satu metode reduksi dimensi yang banyak digunakan adalah *t-distributed Stochastic Neighbor Embedding* (t-SNE). Metode ini memetakan data berdimensi tinggi ke ruang berdimensi rendah dengan mempertahankan kedekatan antar data yang memiliki

karakteristik serupa, sehingga sampel yang mirip cenderung membentuk kelompok (*cluster*) pada hasil visualisasi (Cieslak et al., 2020).

### 3.11. Grad-CAM

*Gradient-weighted Class Activation Mapping* (Grad-CAM) merupakan metode visualisasi yang digunakan untuk menghasilkan penjelasan visual terhadap keputusan model berbasis *Convolutional Neural Network* (CNN). Metode ini menghasilkan peta aktivasi atau *heatmap* yang menunjukkan wilayah pada citra *input* yang memberikan kontribusi terbesar terhadap prediksi suatu kelas. Grad-CAM dikembangkan oleh Selvaraju dkk. (2017) sebagai pengembangan dari *Class Activation Mapping* (CAM) yang memiliki fleksibilitas lebih tinggi karena dapat diterapkan pada berbagai arsitektur CNN tanpa memerlukan perubahan struktur jaringan maupun proses pelatihan ulang (*re-training*). Dengan demikian, Grad-CAM dapat digunakan untuk meningkatkan interpretabilitas model dengan menunjukkan bagian citra yang menjadi dasar pengambilan keputusan model.

Grad-CAM bekerja dengan memanfaatkan informasi gradien dari skor prediksi kelas target terhadap *feature map* pada lapisan konvolusi terakhir. Gradien tersebut digunakan untuk mengukur tingkat kepentingan setiap *feature map* terhadap prediksi kelas tertentu melalui proses *global average pooling*. Bobot kepentingan setiap *feature map* dihitung menggunakan persamaan:

$$\alpha_k^c = \frac{1}{Z}\sum_i \sum_j \frac{\partial y^c}{\partial A_{ij}^k} \tag{3.22}$$

dengan $\alpha_k^c$ merupakan bobot *feature map* ke-k terhadap kelas $c$, $y^c$ adalah skor prediksi sebelum fungsi *softmax* untuk kelas $c$, $A_{ij}^k$ merupakan nilai aktivasi pada posisi (i,j) dari *feature map* ke-k, sedangkan $Z$ menyatakan jumlah elemen pada *feature map*. Bobot tersebut merepresentasikan besarnya kontribusi masing-masing feature map terhadap prediksi kelas target. Setelah bobot setiap *feature map* diperoleh, Grad-CAM menghasilkan peta aktivasi dengan menjumlahkan seluruh *feature map* yang telah dikalikan dengan bobotnya, kemudian menerapkan fungsi *Rectified Linear Unit* (ReLU) untuk mempertahankan kontribusi positif terhadap kelas target. Peta aktivasi Grad-CAM dirumuskan sebagai berikut:

$$L^c_{Grad-CAM} = ReLU\left(\sum_k \alpha_k^c A^k\right) \tag{3.23}$$

dengan $L^c_{Grad-CAM}$ merupakan peta aktivasi untuk kelas $c$, $A^k$ adalah *feature map* ke-k, dan $\alpha_k^c$ merupakan bobot kepentingan yang diperoleh dari Persamaan (3.22). Fungsi ReLU digunakan untuk menghilangkan kontribusi negatif sehingga hanya fitur yang mendukung prediksi kelas target yang dipertahankan. Hasil akhir Grad-CAM berupa *heatmap* yang dapat dioverlay pada citra asli untuk menunjukkan area yang paling berpengaruh terhadap keputusan model. Visualisasi ini membantu meningkatkan transparansi model, mengevaluasi keandalan prediksi, serta memudahkan analisis apakah model benar-benar berfokus pada objek yang relevan dalam proses klasifikasi (Selvaraju et al., 2017).

# BAB IV
# METODOLOGI PENELITIAN

## 4.1. Populasi dan Sampel Penelitian

Populasi dalam penelitian ini adalah seluruh citra wajah digital yang merepresentasikan wajah asli (*real image*) dan wajah *deepfake* (*AI-generated*) dari individu berkewarganegaraan Indonesia yang digunakan sebagai objek penelitian dalam tugas klasifikasi wajah. Populasi tersebut mencakup citra wajah yang dihasilkan secara langsung dari individu nyata serta citra wajah *deepfake* yang dimanipulasi menggunakan teknologi kecerdasan buatan (*Artificial Intelligence*).

Sampel penelitian yang digunakan merupakan keseluruhan *dataset* citra wajah yang telah disiapkan dan digunakan dalam proses pelatihan (*training*) serta pengujian (*testing*) model klasifikasi. Total *dataset* yang digunakan dalam penelitian ini berjumlah 1.000 citra wajah, yang terbagi secara seimbang ke dalam dua kelas yaitu 500 citra wajah asli dan 500 citra wajah *deepfake*.

Sumber data yang digunakan dalam penelitian ini merupakan data primer, karena seluruh citra wajah baik wajah asli maupun wajah *deepfake* dikumpulkan secara mandiri oleh peneliti. Citra wajah asli diperoleh melalui proses pengumpulan data wajah individu, sedangkan citra wajah *deepfake* dihasilkan menggunakan teknologi kecerdasan buatan (*AI-generated images*). Penggunaan data primer ini bertujuan untuk memastikan kesesuaian karakteristik *dataset* dengan kebutuhan penelitian, sehingga hasil penelitian diharapkan dapat memberikan kontribusi yang lebih relevan dan kontekstual dalam bidang deteksi wajah *deepfake*.

## 4.2. Tempat dan Waktu Penelitian

Penelitian ini dilaksanakan secara daring (online) dengan memanfaatkan perangkat komputasi untuk proses pengolahan dan analisis data. Seluruh tahapan penelitian mulai dari pengumpulan *dataset* wajah asli dan pembuatan wajah *deepfake* (*AI-generated*), proses pra-pemrosesan data, pelatihan (*training*) model *Convolutional Neural Network* (CNN) dengan arsitektur EfficientNet-B0, hingga pengujian (*testing*) dan evaluasi kinerja model, dilakukan menggunakan lingkungan komputasi berbasis perangkat lunak.

## 4.3. Definisi Operasional Variabel Penelitian

Variabel yang digunakan dalam penelitian ini ditampilkan dalam Tabel 4.1 tentang penjelasan dan definisi variabel.

**Tabel 4.1.** Definisi Variabel Penelitian

| Variabel | Definisi Variabel Penelitian |
|---|---|
| Citra Wajah Asli | Citra wajah manusia asli (*real face*) orang Indonesia yang diperoleh dari sumber *open source* di internet, tanpa manipulasi digital maupun proses generasi oleh kecerdasan buatan (AI). |
| Citra Wajah *Deepfake* (*AI-generated*) | Citra wajah orang Indonesia yang dihasilkan atau dimanipulasi menggunakan teknologi kecerdasan buatan (*AI-generated*). |

## 4.4. Alat dan Cara Organisir Data

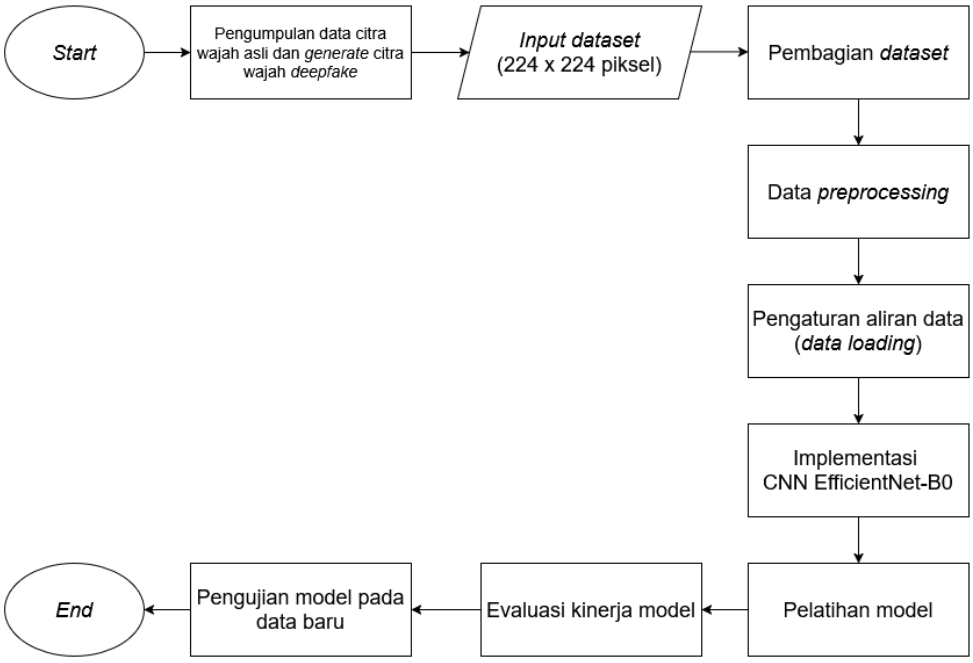
Alat yang digunakan dalam penelitian ini adalah Google Colaboratory (Google Colab) sebagai lingkungan komputasi berbasis *cloud*. Google Colab dipilih karena menyediakan akses terhadap sumber daya komputasi dan *Graphics Processing Unit* (GPU) yang memadai, serta mendukung berbagai pustaka *deep learning* seperti TensorFlow dan Keras yang diperlukan dalam pengembangan model *Convolutional Neural Network* (CNN) dengan arsitektur EfficientNet-B0. Selain itu, Google Colab memungkinkan proses pengolahan data dan pelatihan model dilakukan secara daring tanpa memerlukan instalasi perangkat lunak tambahan pada komputer lokal.

Pengorganisasian data dalam penelitian ini dilakukan dengan menyusun *dataset* citra wajah ke dalam struktur folder yang terpisah antara data latih (*training data*) dan data uji (*testing data*). Masing-masing folder tersebut terdiri dari dua subfolder kelas, yaitu *real image* dan *deepfake image*, yang berisi citra wajah asli dan citra wajah *deepfake*. Struktur organisasi data ini bertujuan untuk memudahkan proses pemanggilan data oleh model CNN serta memastikan proses pelatihan dan pengujian berjalan secara sistematis dan konsisten. Seluruh *dataset* diunggah dan diakses melalui Google Colab, baik secara langsung maupun melalui integrasi dengan layanan penyimpanan *cloud* seperti Google Drive.

## 4.5. Metode Penelitian

Metode penelitian yang digunakan dalam penelitian ini adalah metode eksperimen kuantitatif dengan pendekatan *deep learning*. Metode ini dipilih karena memungkinkan pengujian kinerja model secara sistematis, objektif, dan terukur berdasarkan data numerik berupa citra digital dan metrik evaluasi model. Penelitian ini bertujuan untuk menguji kemampuan model *Convolutional Neural Network* (CNN) dengan arsitektur EfficientNet-B0 dalam mendeteksi dan mengklasifikasikan wajah asli (*real image*) dan wajah palsu hasil manipulasi kecerdasan buatan (*deepfake image*) berdasarkan citra wajah digital.

Pendekatan eksperimen digunakan karena peneliti dapat mengendalikan seluruh proses, mulai dari penyusunan *dataset*, pra-pemrosesan data, pelatihan model, hingga evaluasi kinerja model, sehingga hasil penelitian dapat dianalisis secara objektif dan dapat direplikasi. Tahapan penelitian dalam studi ini terdiri dari beberapa fase utama yang saling berurutan, sebagaimana ditunjukkan pada Gambar 4.1.




**Gambar 4.1.**Tahapan Penelitian

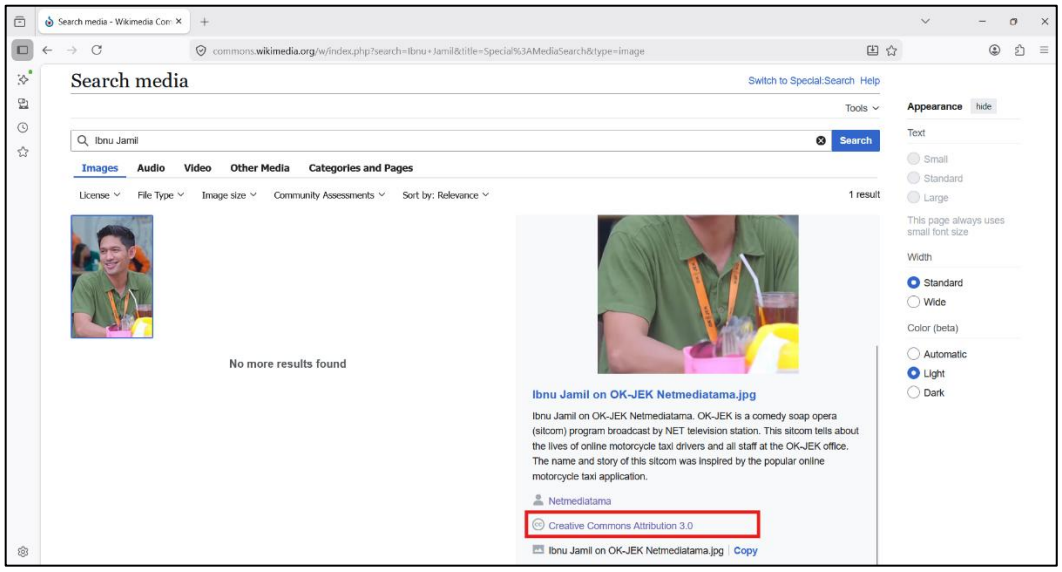
## 4.6. *Dataset* Penelitian

*Dataset* yang digunakan dalam penelitian ini secara spesifik berfokus pada profil wajah individu. Untuk memfasilitasi kebutuhan klasifikasi multikelas pada model *Convolutional Neural Network* (CNN), *dataset* dikategorikan ke dalam

empat kelas utama, yaitu *female real*, *male real*, *female fake*, dan *male fake*. Pengumpulan data dilakukan melalui dua pendekatan, yaitu pengunduhan citra publik untuk kelas asli (*real*) dan generasi citra buatan untuk kelas manipulasi (*fake*).

### 4.6.1. Pengumpulan Citra Wajah Asli

Citra wajah asli untuk kelas *female real* dan *male real* diperoleh dari Wikimedia Commons, yaitu repositori media terbuka yang menyediakan berbagai *file* dengan lisensi yang mengizinkan penggunaan kembali secara bebas untuk keperluan akademik maupun nonkomersial. *Dataset* yang diambil berupa gambar wajah publik yang menampilkan individu berkewarganegaraan Indonesia. Pemilihan sumber ini didasarkan pada ketersediaan citra dengan resolusi yang sesuai untuk proses pelatihan dan evaluasi model serta kejelasan informasi lisensi yang mendukung penggunaan untuk kepentingan penelitian akademik.




**Gambar 4.2.** Unduh Gambar di Wikimedia Commons

### 4.6.2. Generasi Citra Wajah Menggunakan AI

Citra wajah manipulasi untuk kelas *female fake* dan *male fake* tidak diambil dari internet, melainkan disintesis secara mandiri (*self-generated*) menggunakan teknologi *Artificial Intelligence* (AI). Proses generasi ini memanfaatkan *platform* Gemini AI yang ditenagai oleh model AI generatif Nano Banana 2. Model ini dipilih karena kemampuannya dalam menghasilkan gambar tingkat profesional yang *hiper-realistis* dengan detail mikro yang sangat tinggi.

Sebagai bentuk pemenuhan batasan penelitian, setiap instruksi (*prompt*) secara eksplisit menyertakan frasa pengikat demografis yaitu "wanita Indonesia" untuk data *female fake* dan "pria Indonesia" untuk data *male fake*. Penambahan kata kunci tersebut bertujuan untuk memastikan bahwa model generatif mampu merepresentasikan karakteristik rasial serta fitur wajah (fenotip) yang spesifik pada populasi Indonesia. Dengan demikian, citra *deepfake* yang dihasilkan menjadi lebih kontekstual dan relevan dengan studi kasus yang diangkat dalam penelitian ini.

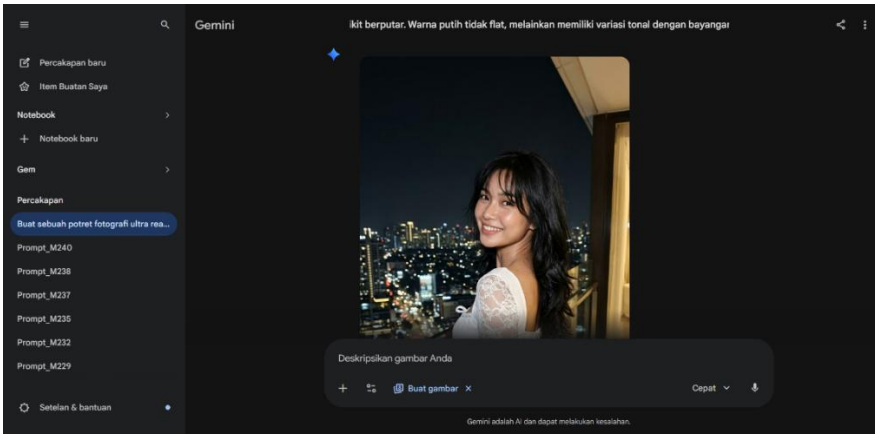
Selain itu, untuk memastikan bahwa citra *deepfake* yang dihasilkan memiliki kualitas yang memadai untuk menguji kinerja arsitektur deteksi, diterapkan metode *prompt engineering* tingkat lanjut. Instruksi (*prompt*) dirancang secara sistematis untuk mendorong model AI menghasilkan tekstur kulit yang lebih autentik, pencahayaan yang alami, serta meminimalkan kemunculan artefak khas AI yang mudah dikenali. Berikut merupakan contoh rancangan *prompt* untuk *generate* data citra *deepfake*:

a. *Prompt Generate* Data Citra *Female Fake*

"*Buat sebuah potret fotografi ultra realistis seorang wanita Indonesia, ditangkap dalam kondisi asli tanpa manipulasi digital, menampilkan karakteristik kulit manusia yang autentik dan detail tingkat tinggi. Permukaan kulit terlihat hidup dengan tekstur alami yaitu pori pori halus yang tampak jelas, variasi kecil pada warna kulit, serta pantulan cahaya yang mengikuti struktur kulit secara realistis. Hindari tampilan sintetis seperti CGI atau efek smoothing berlebihan. Gunakan pencahayaan alami yang lembut dengan dinamika cahaya dan bayangan yang realistis, menyerupai kondisi pemotretan dunia nyata. Difoto menggunakan smartphone flagship dengan simulasi lensa portrait sekitar 85mm, depth of field natural, fokus presisi tinggi pada wajah, dan kualitas gambar setara kamera profesional. Komposisi vertikal rasio 9:16 dengan resolusi tinggi. Buat potret candid yang realistis dari seorang wanita Indonesia yang sangat cantik, berdiri di balkon hotel bintang 5 dengan pemandangan city light yang sangat indah. Ia berpose dengan tubuh menghadap sedikit ke samping menjauhi kamera, lalu menoleh ke belakang ke arah kamera, menciptakan kesan candid seolah baru saja dipanggil. Ekspresinya tersenyum manis dengan tatapan hangat dan natural, memberikan nuansa elegan, santai, dan berkelas. Rambutnya berwarna hitam*

*natural, ditata dalam gaya butterfly cut panjang hingga sekitar pinggang. Potongannya berlapis dengan volume yang jatuh secara alami. Poni curtain bangs tampak tipis dan ringan, sedikit terbelah mengikuti arah putaran kepala. Ujung rambut tampak sedikit feathered dengan tekstur yang tidak sepenuhnya seragam, menambah kesan hidup dan realistis. Ia mengenakan dress putih elegan dengan desain fitted yang membentuk tubuh secara natural seperti pada referensi. Bagian atas dress memiliki struktur corset dengan jahitan vertikal yang tegas namun tetap halus, memberikan siluet ramping yang anggun. Garis leher berbentuk square neckline yang terbuka namun tetap classy, memperlihatkan area leher dan tulang selangka secara elegan yang sedikit terekspos akibat pose menoleh. Pada bagian dada terdapat detail kecil seperti pita tipis di tengah yang memberikan aksen feminin yang subtle. Dress dilengkapi dengan lengan panjang berbahan lace transparan dengan tekstur floral halus, yang mengikuti bentuk tangan dan sedikit melebar di bagian pergelangan dengan efek flowy yang lembut. Material dress terlihat seperti kombinasi lace premium dan kain lembut yang jatuh natural mengikuti tubuh, dengan tekstur detail yang terlihat jelas terutama pada bagian lace. Rok bagian bawah jatuh ringan dengan sedikit flare dan lipatan alami, menciptakan gerakan halus yang mengikuti arah tubuh yang sedikit berputar. Warna putih tidak flat, melainkan memiliki variasi tonal dengan bayangan lembut dan highlight natural sesuai arah cahaya. Tidak ada efek glowing berlebihan, dan detail jahitan terlihat rapi serta realistis. Latar belakang menunjukkan balkon hotel bintang 5 dengan railing kaca transparan yang sedikit memantulkan cahaya kota. Di kejauhan terlihat city lights malam hari dengan pemandangan lampu gedung, jalan, dan kendaraan. Beberapa titik cahaya memiliki variasi warna hangat dan dingin, menciptakan depth yang realistis. Langit malam terlihat gelap dengan sedikit noise alami khas kamera low light. Refleksi halus dari lampu kota tampak pada permukaan kaca dan elemen sekitar. Sudut kamera diambil dari sedikit high angle sekitar setinggi mata atau sedikit di atas, dengan framing dari pinggang ke atas medium close up. Komposisi dibuat sedikit off center dengan arah pandang subjek yang menoleh ke belakang menuju kamera, menciptakan dinamika visual yang natural dan tidak staged. Perspektif wajah tetap proporsional dengan sedikit twist pada leher dan bahu yang terlihat realistis.*

*Simulasi lensa portrait sekitar 85mm dengan depth of field realistis, wajah menjadi titik fokus utama yang sangat tajam terutama pada area mata yang mengarah ke kamera, sementara latar belakang blur secara natural dengan transisi halus. Tidak ada efek cut out. Fokus sedikit breathing seperti kamera asli dengan ketajaman maksimal di area mata. Pencahayaan berasal dari kombinasi ambient city light dan soft warm light dari dalam ruangan hotel yang mengenai sisi wajah dan sebagian punggung secara lembut, mengikuti arah pose menoleh. Cahaya menciptakan highlight natural pada pipi, hidung, dan rahang, serta bayangan halus di sisi wajah lainnya. White balance sedikit hangat seperti kamera smartphone di malam hari. Gaya fotografi menyerupai candid lifestyle photography dengan pendekatan natural light, seperti dokumentasi momen nyata yang elegan. Sedikit imperfection seperti micro motion akibat gerakan menoleh, stray hair, dan subtle noise dipertahankan untuk meningkatkan keaslian. Tampilan menyerupai hasil foto RAW dari smartphone flagship tanpa retouch berlebihan. Detail krusial: tekstur kulit harus tetap realistis dengan pori pori terlihat, garis halus alami, baby hair, dan helai rambut acak. Pencahayaan mengikuti natural falloff dengan bayangan lembut. Kualitas gambar tajam namun tidak overprocessed. Negative Prompt: blur, low detail, soft focus berlebihan, lighting tidak natural, highlight terlalu keras, shadow pecah, kulit plastik, efek beauty filter, proporsi wajah tidak realistis, deformasi anatomi, jari tidak normal, ekspresi kaku, komposisi tidak natural, latar belakang terlalu ramai, watermark, teks, CGI, render 3D, tampilan kartun, oversaturated, overprocessed, artificial sharpness*".

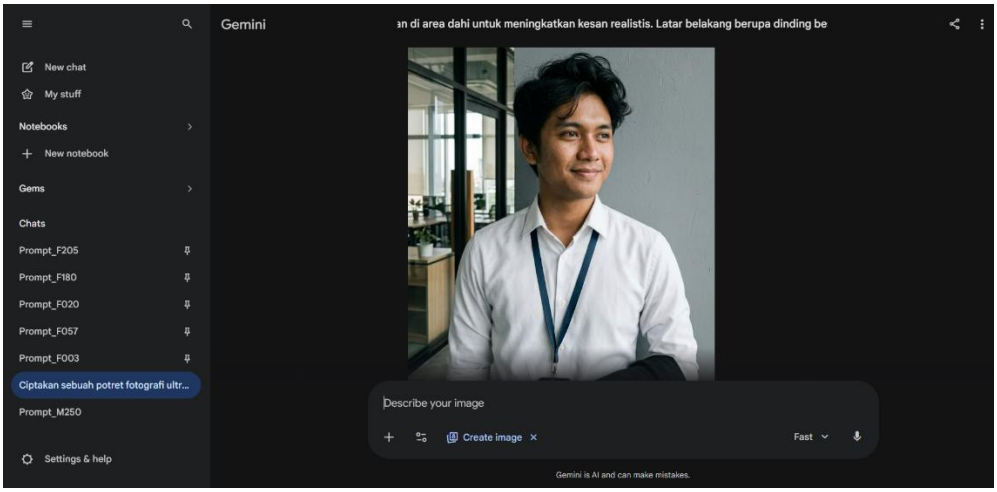



**Gambar 4.3.** *Generate* Data Citra *Female Fake*

b. *Prompt Generate* Data Citra *Male Fake*

"*Ciptakan sebuah potret fotografi ultra-realistis dari seorang pria muda Indonesia, ditangkap seolah menggunakan kamera nyata tanpa manipulasi digital berlebihan. Kulit wajah harus menampilkan tekstur autentik dengan detail realistis, pori-pori halus, variasi warna kulit yang natural, serta pantulan cahaya yang mengikuti struktur wajah secara realistis. Hindari tampilan sintetis seperti CGI, render 3D, atau efek smoothing berlebihan. Gunakan pencahayaan indoor yang lembut dan realistis, menyerupai kondisi kantor modern dengan pencahayaan ambient yang menyebar alami. Simulasikan pengambilan menggunakan kamera smartphone flagship dengan lensa portrait (sekitar 50–85mm), menghasilkan fokus tajam dengan depth natural. Komposisi vertikal rasio 9:16 dengan framing profesional setengah badan. Seorang pria muda Indonesia berdiri dalam pose setengah badan dengan tampilan rapi dan profesional. Ia mengenakan kemeja putih formal dengan lengan digulung hingga bawah siku. Di lehernya tergantung lanyard polos berwarna biru tua tanpa tulisan, jatuh secara natural mengikuti bentuk tubuh. Salah satu tangannya memegang jas hitam yang dilepas, digantung santai di sisi tubuh, menciptakan kesan elegan namun tetap relaxed. Postur tubuh sedikit menghadap samping (3/4 angle), dengan kepala menoleh ke arah samping. Ekspresi wajah menunjukkan senyum lembut yang natural, dengan tatapan mata yang hidup dan tidak kaku. Rambut hitam alami, tebal dan bervolume, ditata rapi dengan sedikit tekstur messy. Beberapa helai rambut jatuh ringan di area dahi untuk meningkatkan kesan realistis. Latar belakang berupa dinding besar berwarna abu-abu dengan tekstur halus yang terlihat jelas dan tajam (tidak blur), mendominasi frame utama. Di sisi samping terlihat area kantor modern dengan elemen kaca transparan berbingkai hitam dan interior minimalis yang clean. Pencahayaan pada dinding menciptakan gradasi cahaya lembut berbentuk oval/spotlight alami, seolah berasal dari lampu indoor yang memantul, memberikan kesan mewah dan modern. Detail lantai dan arsitektur terlihat realistis tanpa distorsi, dengan perspektif yang natural. Detail penting: hasil akhir harus terlihat seperti foto asli dari kamera smartphone, dengan ketajaman natural tanpa overprocessing. Tekstur kulit harus tetap utuh tanpa smoothing berlebihan. Pastikan detail kecil seperti tekstur dinding, pantulan*

*cahaya pada kaca, serta material lantai terlihat realistis dan tajam. Catatan tambahan: fokus pada realisme mikro seperti pori-pori kulit, garis halus wajah, baby hair, serta tekstur kain dan lingkungan sekitar. Pencahayaan harus mengikuti kondisi dunia nyata dengan gradasi bayangan yang halus (natural light falloff), tanpa efek dramatis berlebihan. Hasil akhir menyerupai foto RAW dari kamera asli yang jernih, detail tinggi, dan tidak terlihat artificial. Negative Prompt (larangan): blur, latar belakang bokeh, low detail, resolusi rendah, pencahayaan tidak realistis, highlight terlalu terang, shadow pecah, kulit terlalu halus seperti plastik, efek beauty filter, proporsi tubuh tidak wajar, tangan/jari cacat, ekspresi kaku, pose tidak natural, efek CGI, render 3D, gaya kartun, oversaturated, over-sharpen, watermark, teks*".




**Gambar 4.4.** *Generate* Data Citra *Male Fake*

## 4.7. Pembagian Data

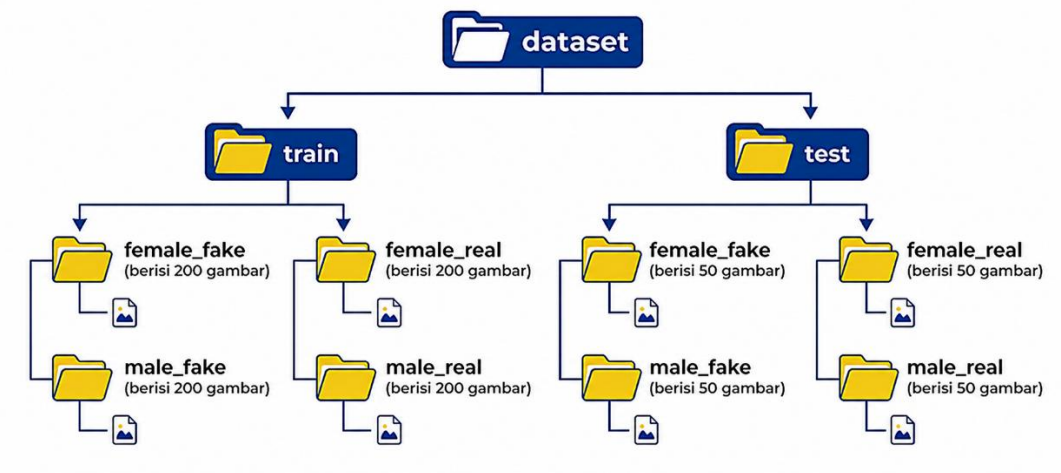
Penelitian ini menggunakan total 1.000 citra wajah yang didistribusikan secara seimbang ke dalam empat kelas target, yaitu *female real*, *female fake*, *male real*, dan *male fake*. Guna menghindari bias evaluasi dan memastikan model memiliki kemampuan generalisasi yang optimal, populasi data dipartisi ke dalam tiga *subset* utama yaitu data latih (*training set*), data validasi (*validation set*), dan data uji (*testing set*). Proses pemisahan ini diimplementasikan melalui mekanisme dua tahap, yakni partisi manual berbasis direktori dan partisi terprogram berbasis skrip komputasional.

### 4.7.1. Pembagian Data Latih dan Data Uji

Pada tahap awal, *dataset* dibagi menjadi dua bagian utama, yaitu data latih (*training set*) dan data uji (*testing set*) dengan rasio 80:20. Pembagian ini berarti

80% dari total data digunakan sebagai data latih dan 20% sisanya digunakan sebagai data uji. Rasio 80:20 merupakan salah satu rasio pembagian data yang banyak digunakan dalam penelitian *machine learning* karena menyediakan jumlah data latih yang memadai untuk proses pembelajaran model sekaligus mempertahankan data uji yang cukup untuk mengevaluasi performa model. Sivakumar et al. (2024) menyatakan bahwa rasio tersebut merupakan salah satu rasio yang umum digunakan dalam literatur. Selain itu, Nguyen et al. (2021) menggunakan rasio 80:20 sebagai salah satu skenario pembagian data dalam mengevaluasi pengaruh rasio data latih dan data uji terhadap performa model.

Pada penelitian ini, *dataset* yang digunakan telah disusun dan dipisahkan sejak awal secara manual ke dalam dua *folder* utama yaitu *train* dan *test*. Pembagian awal ini bertujuan untuk memastikan bahwa data uji sama sekali tidak terlibat dalam proses pelatihan model, sehingga evaluasi kinerja dapat dilakukan secara objektif dan mencegah terjadinya kebocoran data (*data leakage*). Data latih terdiri dari 800 citra wajah, sedangkan data uji terdiri dari 200 citra wajah. Pada data latih, masing-masing dari keempat kelas tersebut secara merata diisi dengan 200 citra wajah. Sementara itu, pada data uji masing-masing kelas terdiri dari 50 citra wajah.




**Gambar 4.5.** Stuktur Folder *Dataset*

### 4.7.2. Pemisahan Data Validasi

Tahap selanjutnya adalah membagi 800 gambar yang ada di *folder train* menjadi data latih dan data validasi. Proses ini dilakukan secara otomatis menggunakan fungsi `train_test_split` dari *library* `scikit-learn`. Pembagian menggunakan rasio 90:10 (90% untuk data latih dan 10% untuk data validasi)

secara proporsional. Parameter `random_state` juga diatur pada angka tertentu agar pengacakan data selalu menghasilkan urutan yang sama jika kode dijalankan ulang.

Selain itu, untuk mempercepat proses membaca data di Google Colab, program tidak menggandakan *file* gambar secara fisik. Sebagai gantinya, program membuat pintasan (*symlinks*) yang mengarah ke gambar asli di dalam *folder* penyimpanan virtual. Cara ini sangat menghemat waktu pemuatan data tanpa merusak susunan *file* aslinya.

Secara keseluruhan, skenario pembagian dua tahap ini menghasilkan persentase akhir 72% data latih, 8% data validasi, dan 20% data uji. Data validasi (8%) ini sangat penting karena berfungsi sebagai bahan ujian sementara bagi model di setiap putaran pelatihan (*epoch*) untuk memantau performa dan mencegah model terlalu menghafal data latih (*overfitting*). Rincian jumlah gambar untuk masing-masing kelas dapat dilihat pada Tabel 4.2.

**Tabel 4.2.** Distribusi Alokasi *Dataset* pada Masing-Masing Kelas

| Kelas Target | Data Latih (*Train*) | Data Validasi (*Validation*) | Data Uji (*Test*) | Total Keseluruhan |
|---|---|---|---|---|
| *Female Real* | 180 | 20 | 50 | 250 |
| *Female Fake* | 180 | 20 | 50 | 250 |
| *Male Real* | 180 | 20 | 50 | 250 |
| *Male Fake* | 180 | 20 | 50 | 250 |
| Total Populasi | 720 (72%) | 80 (8%) | 200 (20%) | 1.000 (100%) |

## 4.8. *Data Preprocessing* dan *Data Loading*

Tahap pra-pemrosesan data atau *data preprocessing* bertujuan untuk mengondisikan citra mentah agar sesuai dengan format standar masukan arsitektur EfficientNet-B0 serta meningkatkan kualitas pembelajaran model. Langkah pertama yang dilakukan adalah penyesuaian dimensi secara manual, di mana resolusi seluruh gambar diseragamkan menjadi 224 x 224 piksel agar matriks citra dapat dikomputasi dengan baik oleh lapisan konvolusi. Setelah dimensi disesuaikan, data citra melalui tahapan transformasi mencakup normalisasi dan augmentasi. Normalisasi digunakan untuk mengonversi nilai piksel ke dalam rentang tensor yang seragam berdasarkan nilai rata-rata (*mean*) dan standar deviasi standar untuk mempercepat konvergensi. Sementara itu, teknik augmentasi data

diterapkan secara khusus pada himpunan data latih guna memperbanyak variasi visual dan mencegah model terjebak pada penghafalan pola (*overfitting*).

Setelah seluruh data ditransformasi, langkah terakhir adalah mengatur aliran data (*data loading*) sebelum dimasukkan ke dalam model pembelajaran. Proses ini dilakukan menggunakan modul *DataLoader* yang bertugas mengelompokkan data ke dalam ukuran tumpukan komputasi tertentu (*batch size*) dan mengacak urutan citra (*shuffle*) secara otomatis pada setiap putaran iterasi (*epoch*). Pengaturan aliran data ini bertujuan untuk menjaga efisiensi penggunaan memori selama proses pelatihan serta mempertahankan stabilitas proses pembaruan bobot pada jaringan saraf tiruan.

### 4.9. Implementasi EfficientNet-B0

Implementasi arsitektur EfficientNet-B0 dilakukan dengan menerapkan metode *transfer learning* melalui pemuatan bobot *pre-trained* yang telah dilatih pada *dataset* ImageNet. Pendekatan ini dipilih untuk memanfaatkan kemampuan ekstraksi fitur visual yang telah mapan sehingga proses pelatihan menjadi lebih efisien meskipun dengan *dataset* yang lebih spesifik. Tahap modifikasi difokuskan pada bagian kepala klasifikasi (*classifier head*) atau lapisan *fully connected* terakhir, di mana jumlah neuron keluaran disesuaikan menjadi empat unit. Penyesuaian ini bertujuan agar model mampu memetakan fitur-fitur yang telah diekstraksi ke dalam empat kategori target penelitian, yaitu *female fake, female real, male fake,* dan *male real*.

Proses pemelajaran model dikendalikan melalui konfigurasi fungsi kerugian (*loss function*), optimasi (*optimizer*), dan pengaturan laju pemelajaran (*learning rate scheduler*) yang sistematis. Fungsi *Cross-Entropy Loss* didefinisikan sebagai metrik evaluasi kesalahan untuk menghitung selisih antara distribusi probabilitas prediksi dan label aktual dalam klasifikasi multikelas. Pembaruan parameter bobot secara iteratif dijalankan menggunakan *optimizer* Adam yang memiliki keunggulan dalam penyesuaian laju pemelajaran secara adaptif untuk setiap parameter. Guna memastikan model mencapai titik konvergensi yang stabil, diterapkan pula *learning rate scheduler* yang berfungsi untuk menurunkan nilai laju pemelajaran secara bertahap pada interval tertentu, sehingga proses optimasi dapat berlangsung secara lebih presisi dan menghindari terjadinya osilasi pada akhir tahapan pelatihan.

## 4.10. Pelatihan Model

Tahap pelatihan model dilaksanakan melalui mekanisme perulangan (*training loop*) yang memproses seluruh data latih dalam sejumlah *epoch* yang telah ditentukan. Pada setiap iterasi, algoritma melakukan rambatan maju (*forward pass*) di mana tumpukan citra masukan dialirkan ke dalam arsitektur EfficientNet-B0 untuk menghasilkan nilai probabilitas prediksi kelas target. Nilai prediksi keluaran tersebut kemudian dikomparasikan dengan label kelas aktual menggunakan fungsi kerugian (*loss function*) untuk mengalkulasi tingkat komputasi kesalahan model. Berdasarkan nilai kesalahan ini, jaringan saraf tiruan melakukan mekanisme rambatan balik (*backward pass* atau *backpropagation*) guna menghitung nilai turunan gradien terhadap seluruh bobot parameter jaringan. Rangkaian ini ditutup dengan langkah pembaruan bobot (*weight update*) oleh algoritma optimasi, yang bertujuan untuk meminimalkan nilai kerugian pada putaran iterasi selanjutnya.

Setelah siklus pelatihan pada setiap *epoch* selesai dieksekusi, proses secara otomatis dilanjutkan dengan fase validasi untuk mengukur kemampuan generalisasi model. Pada fase evaluasi ini, komputasi gradien dinonaktifkan secara terprogram untuk menghemat alokasi memori komputasi dan mencegah model mempelajari data pengujian. Model ditugaskan untuk melakukan prediksi terhadap kumpulan data validasi guna memantau pergerakan metrik evaluasi utama secara *real-time*, yakni tingkat akurasi dan nilai kerugian validasi (*validation loss*). Mekanisme evaluasi berkelanjutan antara fase pelatihan dan validasi dalam setiap iterasi berperan penting untuk memantau potensi *overfitting* serta memastikan bahwa model mempelajari pola pembeda wajah secara umum, bukan hanya karakteristik spesifik pada data latih.

## 4.11. Evaluasi Model pada Data Uji

Tahap evaluasi pada data uji dilakukan untuk mengukur kinerja model serta kemampuan generalisasi arsitektur pada data yang tidak digunakan selama pelatihan. Evaluasi kuantitatif diawali dengan *confusion matrix* yang membandingkan label aktual (*ground truth*) dan label prediksi model guna mengidentifikasi tingkat keberhasilan serta pola kesalahan klasifikasi pada setiap kelas. Selanjutnya, analisis dilengkapi dengan *classification report* yang memuat metrik *precision*, *recall*, dan *F1-score*. Rangkaian metrik ini memberikan gambaran

mengenai ketepatan dan sensitivitas model dalam mengenali setiap kategori wajah, serta mendukung interpretasi terhadap nilai akurasi keseluruhan secara lebih rinci.

Selain evaluasi berbasis ketepatan label, analisis kinerja juga dilakukan terhadap probabilitas prediksi dan kemampuan pemisahan kelas model. Pengukuran ini menggunakan kurva *Receiver Operating Characteristic* (ROC) dan skor *Area Under Curve* (AUC) dengan pendekatan *One-vs-Rest* (OvR) pada skenario klasifikasi multikelas. Kurva ROC memvisualisasikan hubungan antara *True Positive Rate* (TPR) dan *False Positive Rate* (FPR), sedangkan nilai AUC digunakan untuk menilai kemampuan model dalam membedakan antar kelas. Evaluasi ini dilengkapi dengan analisis distribusi probabilitas prediksi yang dihasilkan oleh fungsi aktivasi *Softmax* pada lapisan akhir model. Analisis distribusi tersebut bertujuan untuk meninjau tingkat keyakinan model terhadap hasil klasifikasi serta mengidentifikasi kemungkinan ambiguitas pada proses pengambilan keputusan.

Untuk melengkapi evaluasi statistik dengan pendekatan *Explainable Artificial Intelligence* (XAI), dilakukan analisis ruang representasi fitur menggunakan algoritma *t-Distributed Stochastic Neighbor Embedding* (t-SNE). Metode reduksi dimensi ini memproyeksikan fitur berdimensi tinggi yang diekstraksi oleh EfficientNet-B0 ke dalam representasi dua dimensi. Melalui visualisasi t-SNE, kedekatan antar-citra dapat diamati untuk melihat pola pengelompokan data berdasarkan kemiripan fitur. Visualisasi ini digunakan untuk meninjau sejauh mana model mampu memisahkan representasi wajah asli dan *deepfake* ke dalam klaster yang berbeda. Pola pengelompokan tersebut memberikan indikasi bahwa model mempelajari karakteristik fitur yang relevan dalam membedakan antar kelas.

### 4.12. Evaluasi Model pada Data Baru

Tahap akhir pengujian dilakukan menggunakan data baru (*in-the-wild data*) yang berada di luar distribusi *dataset* pelatihan, validasi, dan data uji. Evaluasi ini bertujuan untuk mengukur kemampuan generalisasi dan *robustness* model pada berbagai kondisi citra nyata seperti variasi pencahayaan, pose, dan latar belakang yang berbeda. Melalui pengujian pada data baru tersebut, dilakukan peninjauan terhadap kemampuan model dalam mengenali pola umum yang membedakan wajah

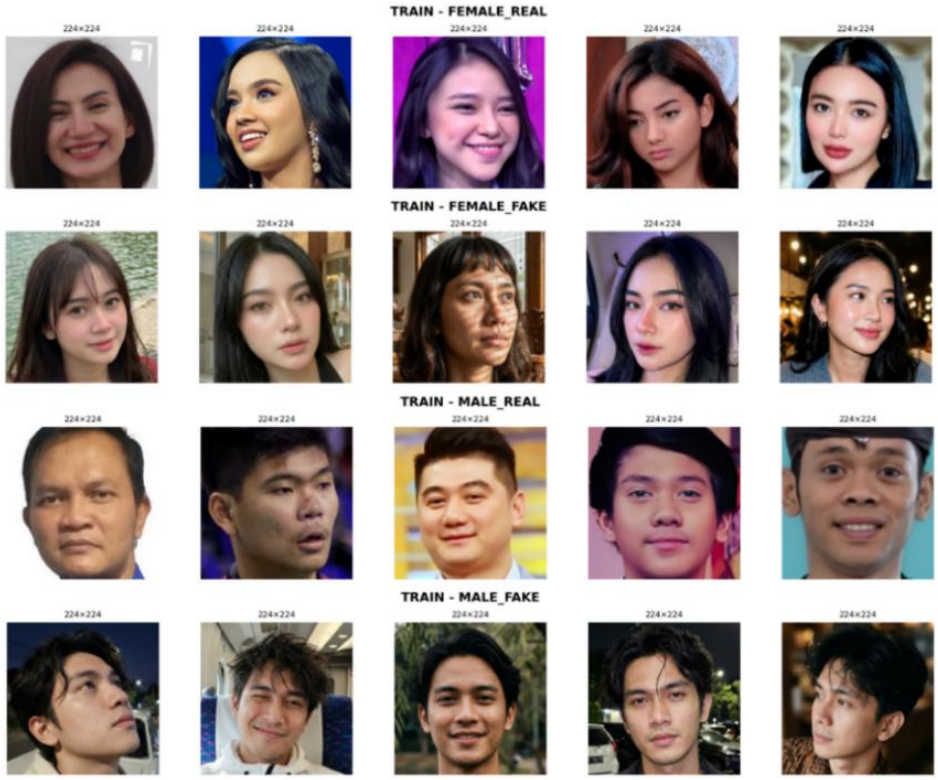
asli dan *deepfake*. Selain itu, pengujian ini juga dilakukan untuk memastikan bahwa kinerja model tidak hanya bergantung pada karakteristik spesifik dari *dataset* pelatihan.

Untuk mendukung interpretasi terhadap proses pengambilan keputusan model, diterapkan metode *Gradient-weighted Class Activation Mapping* (Grad-CAM). Metode ini memanfaatkan informasi gradien pada lapisan konvolusi terakhir untuk menghasilkan visualisasi berupa *heatmap* yang menunjukkan area citra dengan kontribusi terbesar terhadap prediksi model. Melalui visualisasi Grad-CAM, proses klasifikasi pada model dapat dianalisis secara lebih transparan. Pendekatan ini digunakan untuk meninjau apakah model memfokuskan perhatian pada area wajah yang relevan, serta untuk mengamati sejauh mana elemen non-wajah memengaruhi hasil prediksi.

# BAB V
# PEMBAHASAN

## 5.1. Eksplorasi Data

Sebelum memasuki tahap pelatihan model, dilakukan eksplorasi data citra untuk memahami karakteristik *dataset* dan memastikan kesesuaian label kelas. Untuk itu, dilakukan inspeksi visual terhadap sampel acak citra mentah dari *dataset* latih sebagaimana ditunjukkan pada Gambar 5.1. Cuplikan ini diatur secara terstruktur menjadi empat baris yang masing-masing mewakili empat kelas target, yaitu `Train-Female_Real` (wajah asli wanita), `Train-Female_Fake` (wajah rekayasa wanita), `Train-Male_Real` (wajah asli pria), dan `Train-Male_Fake` (wajah rekayasa pria).




**Gambar 5.1.** Sampel Acak Sebelum Transformasi

Pengamatan visual terhadap citra mentah menunjukkan adanya variasi data, meliputi perbedaan ekspresi wajah, sudut pengambilan gambar, kondisi pencahayaan, serta latar belakang. Variasi ini mengindikasikan bahwa *dataset* mencakup berbagai kondisi yang merepresentasikan situasi nyata. Informasi

resolusi awal ditampilkan pada setiap citra, yang memberikan gambaran ukuran sebelum tahap pemrosesan. Seluruh citra kemudian diseragamkan menjadi 224×224 piksel melalui proses resize manual sebelum tahap komputasi. Proses ini dilakukan untuk memastikan area wajah menjadi fokus utama dalam analisis serta meminimalkan pengaruh elemen latar belakang atau bagian tubuh lainnya, sesuai dengan batasan penelitian. Visualisasi awal ini menunjukkan bahwa perbedaan antara citra wajah asli dan *deepfake* sering kali sangat sulit dibedakan oleh mata manusia biasa, sehingga diperlukan pendekatan berbasis *machine learning* untuk mengenali pola halus yang tidak terlihat secara kasat mata.

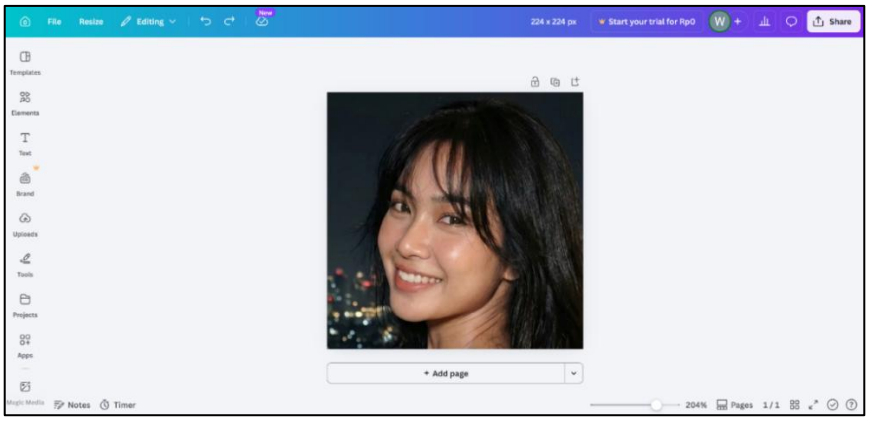
## 5.2. *Data Preprocessing* dan *Data Loading*

Pra-pemrosesan data (*data preprocessing*) merupakan tahapan penting untuk menyesuaikan format citra dengan kebutuhan arsitektur model serta mendukung kemampuan generalisasi selama proses pelatihan. Pada penelitian ini, pra-pemrosesan dilakukan melalui penyesuaian dimensi citra secara manual dan penerapan transformasi data, yang kemudian dilanjutkan dengan pengaturan aliran data (*data loading*).

### 5.2.1. Penyesuaian Dimensi Citra

Sebelum memasuki tahap komputasi, seluruh citra wajah asli dan *deepfake* disesuaikan ukurannya (*resize*) serta dipotong (*crop*) secara manual menggunakan perangkat lunak editor grafis Canva. Pendekatan manual ini sengaja dipilih untuk memastikan bahwa pemotongan gambar terpusat secara presisi pada area wajah, sejalan dengan batasan masalah penelitian. Jika pemotongan dilakukan secara otomatis melalui pustaka kode program, terdapat risiko pemotongan acak yang berpotensi membuang area wajah dan justru menyisakan bagian latar belakang atau bagian tubuh lain.

Melalui proses manual ini, setiap citra dipastikan memiliki resolusi tepat 224 x 224 piksel dengan area target wajah yang proporsional. Standarisasi ukuran dan fokus objek ini merupakan syarat mutlak agar fitur wajah dapat diekstraksi secara optimal oleh lapisan masukan (*input layer*) pada arsitektur EfficientNet-B0. Setelah penyesuaian dimensi dan fokus selesai, citra-citra tersebut didistribusikan ke dalam direktori data latih dan data uji sesuai skenario yang telah ditetapkan.




**Gambar 5.2.** Contoh *Resize* Gambar di Canva

### 5.2.2. Transformasi dan Augmentasi Data

Pada saat proses komputasi berjalan, data citra yang telah disesuaikan ukurannya akan melewati tahap transformasi lanjutan menggunakan pustaka `torchvision.transforms`. Skenario transformasi ini dibedakan antara *subset* data latih dan *subset* data evaluasi (validasi dan uji). Pada data latih, sistem menerapkan teknik augmentasi data untuk memperkaya variasi visual. Langkah ini sangat penting agar model dapat belajar mengenali fitur wajah secara umum dan tidak sekadar menghafal gambar yang sudah ada (*overfitting*). Teknik augmentasi yang diaplikasikan meliputi:

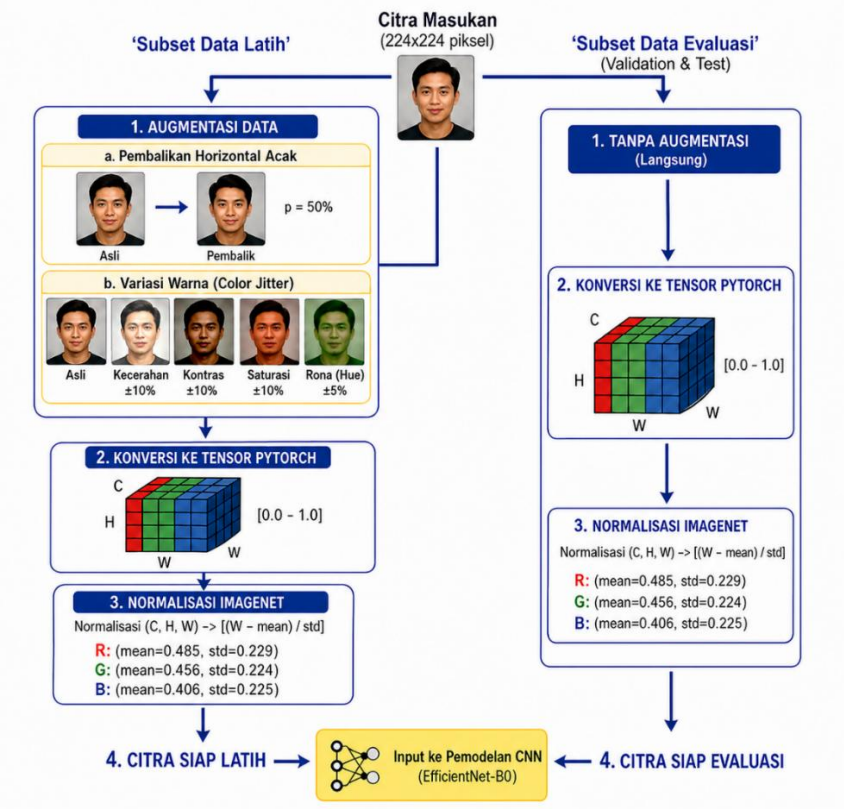
a. Pembalikan Horizontal Acak (*Random Horizontal Flip*): Membalikkan citra secara horizontal (kiri ke kanan) dengan probabilitas 50%. Teknik ini bertujuan untuk mensimulasikan variasi arah hadap wajah subjek.

b. Variasi Warna (*Color Jitter*): Mengacak elemen warna dengan mengubah nilai kecerahan (*brightness*), kontras (*contrast*), dan saturasi (*saturation*) masing-masing sebesar ±10%, serta menggeser rona warna (*hue*) sebesar ±5%. Modifikasi ini melatih model agar lebih tangguh terhadap berbagai perubahan kondisi pencahayaan dan kualitas kamera.

Setelah augmentasi, citra data latih diubah strukturnya menjadi Tensor agar dapat diolah secara matematis oleh kerangka kerja PyTorch. Melalui proses ini, nilai matriks piksel yang semula berada pada rentang 0 hingga 255 diskalakan menjadi rentang 0.0 hingga 1.0. Sebaliknya, pada data validasi dan data uji, teknik augmentasi visual (seperti pembalikan arah dan pengacakan warna) sama sekali tidak diterapkan. Hal ini dilakukan untuk menjaga keaslian data evaluasi, sehingga

pengujian kinerja model benar-benar mencerminkan kondisi data nyata tanpa distorsi. Data evaluasi hanya melewati proses konversi menjadi Tensor.

Sebagai langkah akhir untuk seluruh *subset* data (latih, validasi, dan uji), dilakukan proses normalisasi matriks warna menggunakan nilai rata-rata (mean = [0.485, 0.456, 0.406]) dan standar deviasi (std = [0.229, 0.224, 0.225]) dari standar dataset ImageNet. Normalisasi ini berfungsi untuk memusatkan distribusi nilai piksel, yang pada akhirnya akan mempercepat dan menstabilkan proses pembaruan bobot model (*convergence*) selama pelatihan berlangsung.

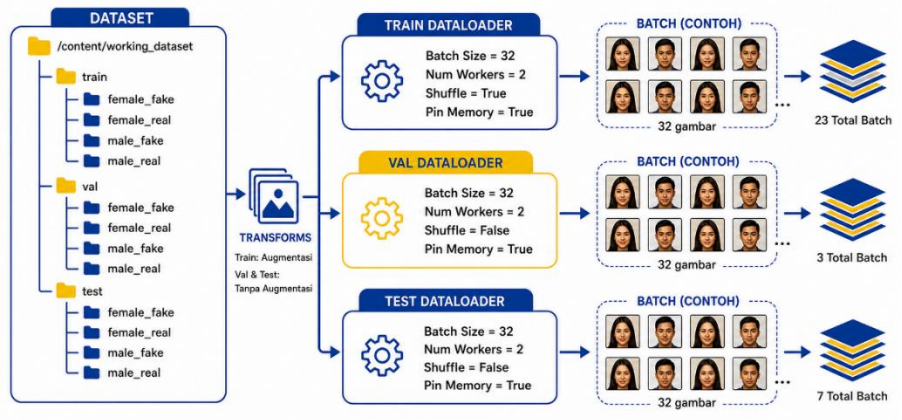



**Gambar 5.3.** Transformasi dan Augmentasi Data

### 5.2.3. Pengaturan Aliran Data (*Data Loading*)

Setelah seluruh citra melewati tahap transformasi, aliran data diatur menggunakan modul `DataLoader` dari kerangka kerja PyTorch. Tahapan ini bertujuan untuk mengelompokkan data dan mendistribusikannya ke dalam model secara efisien selama proses komputasi berlangsung. Untuk mengoptimalkan penggunaan memori pada perangkat keras akselerator (GPU), citra-citra tersebut tidak dimasukkan satu per satu atau sekaligus secara keseluruhan, melainkan dikelompokkan ke dalam tumpukan (*batch*). Pada penelitian ini, ukuran tumpukan

(*batch size*) ditetapkan sebanyak 32 gambar per iterasi. Pemilihan ukuran ini dinilai paling stabil untuk menjaga keseimbangan antara kecepatan pembaruan bobot model dan ketersediaan memori GPU di lingkungan Google Colab. Berdasarkan parameter tersebut, keseluruhan *dataset* secara otomatis terpartisi menjadi beberapa iterasi. Populasi data latih sebanyak 720 citra menghasilkan 23 *batch*, data validasi sebanyak 80 citra terbagi menjadi 3 *batch*, dan data uji sebanyak 200 citra terdistribusi ke dalam 7 *batch*. Sisa pembagian citra pada iterasi terakhir di setiap *subset* tetap diakomodasi dan diproses oleh sistem.




**Gambar 5.4.** Pengaturan Aliran Data

Dalam pelaksanaannya, proses pemuatan data ini dibantu oleh dua inti prosesor (CPU) yang bekerja secara paralel `(num_workers = 2)`. Sistem juga mengaktifkan fungsi `pin_memory` untuk mempercepat transfer matriks data dari memori utama (RAM) menuju memori kartu grafis (GPU), sehingga meminimalkan waktu jeda (*bottleneck*) saat model sedang memproses gambar.

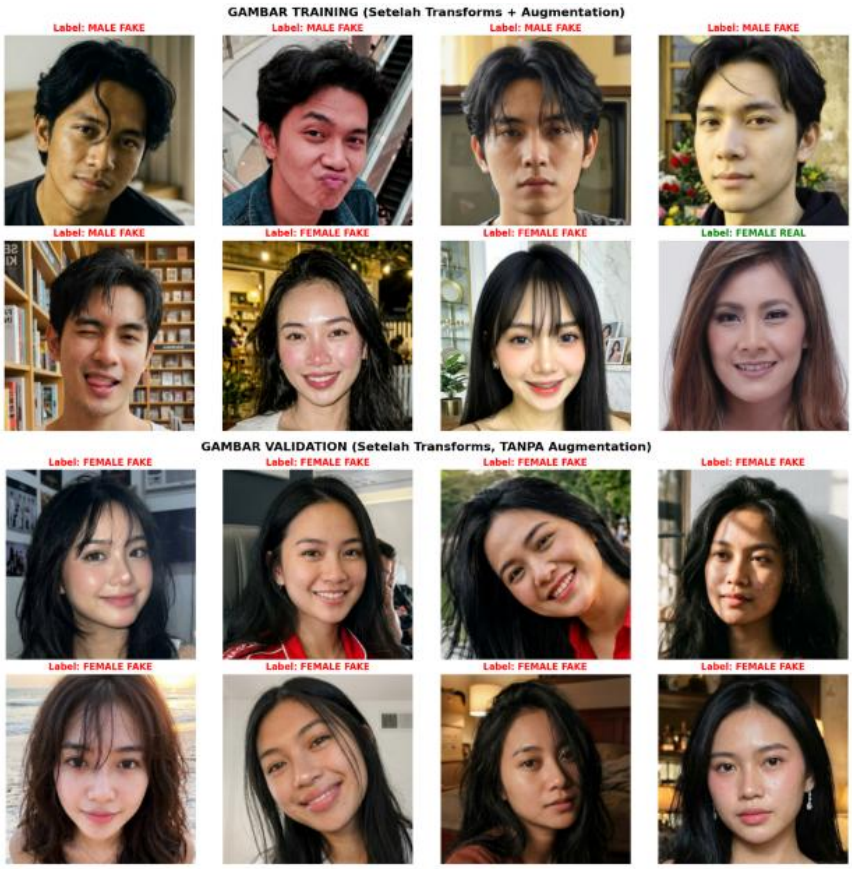
Skenario aliran data ini juga menerapkan perlakuan yang berbeda antara data latih dan data evaluasi. Khusus untuk data latih, urutan gambar selalu diacak (*shuffle*) secara otomatis pada setiap putaran pelatihan (*epoch*). Pengacakan urutan ini krusial untuk memastikan bahwa model benar-benar belajar mengenali pola fitur wajah, bukan sekadar menghafal urutan data yang masuk. Sebaliknya, urutan aliran gambar pada data validasi dan data uji dibiarkan tetap (tidak diacak). Hal ini dilakukan agar proses pengukuran akurasi dan fungsi kerugian (*loss*) dapat dilakukan secara konsisten dan evaluasi yang dihasilkan bersifat objektif.

### 5.2.4. Visualisasi Gambar yang Telah Ditransformasi

Visualisasi citra setelah melalui tahap transformasi ditampilkan pada Gambar 5.5. Visualisasi ini dibagi menjadi dua bagian untuk menunjukkan perbedaan

pemrosesan antara data latih dan data validasi. Bagian atas menampilkan data latih yang telah dikenai transformasi dasar serta teknik augmentasi. Augmentasi bertujuan untuk meningkatkan variasi data sehingga model dapat belajar dari beragam kondisi. Salah satu bentuk augmentasi yang diterapkan adalah *horizontal flip*, yang menghasilkan variasi orientasi wajah dan membantu model mengenali fitur tanpa bergantung pada posisi tertentu.

Selain variasi posisi, diterapkan pula teknik *color jitter* yang secara acak mengubah parameter kecerahan (*brightness*), kontras (*contrast*), saturasi (*saturation*), dan rona (*hue*). Teknik ini mensimulasikan variasi kondisi pencahayaan saat pengambilan citra, sehingga model dapat beradaptasi terhadap perbedaan kualitas pencahayaan. Selain itu, visualisasi warna pada citra data latih terlihat sedikit berbeda (lebih standar) yang disebabkan oleh proses normalisasi. Hal ini penting agar model tidak sensitif terhadap variasi kecerahan warna yang ekstrim.




**Gambar 5.5.** Data Citra Setelah Transformasi

Bagian bawah menampilkan data validasi, di mana hanya diterapkan transformasi dasar tanpa augmentasi. Transformasi ini bertujuan untuk

menyeragamkan ukuran dan bentuk citra. Hasil visual menunjukkan citra yang lebih konsisten tanpa perubahan orientasi seperti *horizontal flip*. Selain itu, teknik *color jitter* tidak diterapkan agar karakteristik warna, kecerahan, dan kontras tetap sesuai dengan kondisi asli. Pendekatan ini sejalan dengan tujuan data validasi, yaitu mengevaluasi kinerja model pada data yang merepresentasikan kondisi umum tanpa modifikasi tambahan. Kedua visualisasi juga menampilkan label kelas pada setiap citra, yang menunjukkan bahwa kesesuaian antara data dan label tetap terjaga setelah melewati alur pemrosesan data.

### 5.2.5. Perbandingan Data Sebelum dan Sesudah Transformasi

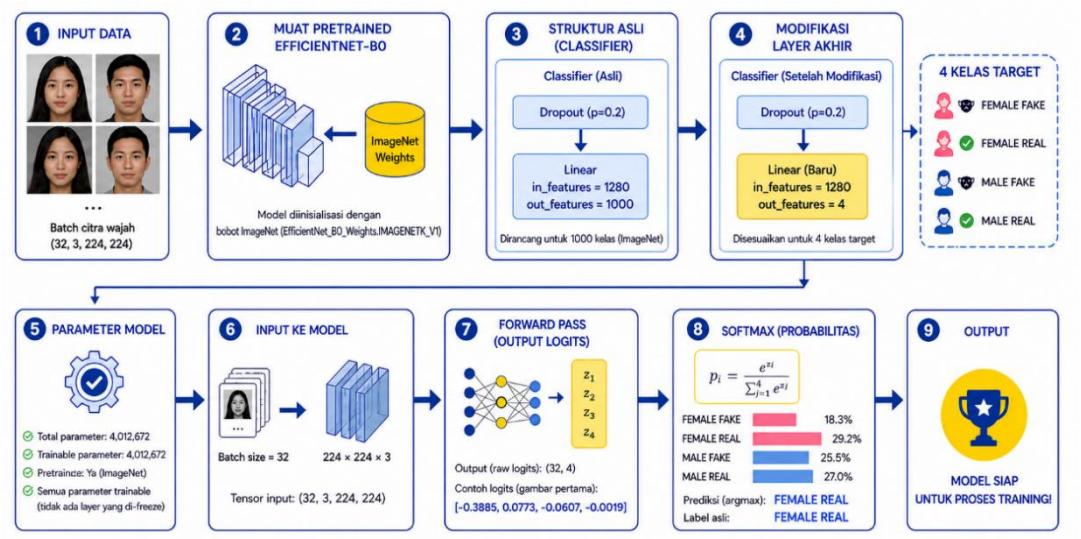
Perbandingan langsung antara citra mentah sebelum transformasi dan citra hasil transformasi menunjukkan perubahan substansial yang bertujuan untuk mengoptimalkan proses pelatihan model EfficientNet-B0. Pertama, terdapat standarisasi bentuk dan ukuran. Inspeksi visual awal pada Gambar 5.1 menunjukkan resolusi mentah yang mungkin terdapat sedikit variasi secara internal meskipun tertulis 224x224 piksel, sedangkan Gambar 5.5 menunjukkan bahwa semua citra secara nyata telah disesuaikan menjadi bentuk persegi sempurna yang seragam. Langkah standarisasi ini sangat penting karena arsitektur jaringan saraf konvolusi membutuhkan ukuran masukan yang tetap agar perhitungan matematis pada lapisan jarigan dapat berjalan dengan stabil.

Kedua, terdapat perubahan visual pada warna dan pencahayaan yang disebabkan oleh proses normalisasi. Citra mentah pada Gambar 5.1 memiliki saturasi warna yang lebih tinggi dan variasi kecerahan yang asli, sedangkan citra pada Gambar 5.5 terlihat sedikit lebih datar atau standar. Normalisasi ini penting agar model dapat fokus mempelajari pola fitur wajah seperti kerutan atau bayangan halus, tanpa terdistraksi oleh variasi kecerahan atau kontras warna yang ekstrim antar gambar yang berbeda.

Terakhir, penerapan augmentasi pada data latih di Gambar 5.5 menciptakan variasi spasial (gambar terbalik) serta variasi rona melalui teknik *color jitter* yang tidak ditemukan pada inspeksi visual awal di Gambar 5.1. Langkah perbandingan ini memastikan bahwa seluruh tahapan pemrosesan data, mulai dari prapemrosesan dasar hingga augmentasi telah berjalan sesuai rancangan komputasi dan siap untuk digunakan dalam fase pelatihan model.

## 5.3. Implementasi EfficientNet-B0

Penelitian ini menggunakan arsitektur Convolutional Neural Network (CNN) berbasis EfficientNet-B0. Arsitektur ini diimplementasikan menggunakan pendekatan *transfer learning*, di mana model diinisialisasi dengan memuat bobot terlatih sebelumnya (*pre-trained weights*) dari *dataset* ImageNet (`EfficientNet_B0_Weights.IMAGENET1K_V1`). Penggunaan bobot dasar ini memungkinkan model untuk memanfaatkan kemampuan ekstraksi fitur visual yang telah mapan, sehingga mempercepat proses konvergensi saat dilatih ulang pada *dataset* wajah. Pada penelitian ini, proses adaptasi arsitektur sistem dilakukan melalui serangkaian tahapan prosedural yaitu inspeksi dan modifikasi lapisan klasifikasi (*classifier layer*), penentuan status parameter pelatihan, serta uji coba aliran maju (*forward pass*) dan fungsi aktivasi.




**Gambar 5.6.** Alur Kerja Arsitektur Sistem

### 5.3.1. Modifikasi Lapisan Klasifikasi

Arsitektur bawaan EfficientNet-B0 dirancang secara spesifik untuk memecahkan tugas klasifikasi citra berskala besar yang mencakup 1.000 kategori objek, sebagaimana standar pada *dataset* ImageNet. Pada bagian akhir dari arsitektur ini, proses ekstraksi fitur yang dilakukan oleh jaringan lapisan konvolusi (*convolutional base*) akan menghasilkan matriks yang kemudian direpresentasikan sebagai 1.280 fitur masukan (`in_features`). Kumpulan fitur tingkat tinggi ini selanjutnya diteruskan ke dalam modul klasifikasi akhir (*classifier module*) yang

terstruktur sebagai fungsi *sequential* untuk melakukan pemetaan probabilitas terhadap 1.000 kategori keluaran tersebut.

Secara anatomis, modul klasifikasi bawaan ini terdiri dari dua komponen utama, yaitu lapisan *dropout* dan lapisan *linear* (*fully connected layer*). Lapisan *dropout* pada arsitektur ini diimplementasikan dengan nilai probabilitas 0,2 (p=0.2), yang bertindak sebagai teknik regularisasi jaringan. Mekanisme ini bekerja dengan cara memutuskan koneksi (menonaktifkan) 20% neuron secara acak pada setiap iterasi pelatihan. Penerapan *dropout* ini secara teoretis berfungsi untuk mencegah model terlalu bergantung pada jalur fitur tertentu, sehingga secara efektif mereduksi risiko model sekadar menghafal data (*overfitting*) dan meningkatkan kemampuan generalisasinya.

Guna menyelaraskan arsitektur dengan batasan ruang lingkup penelitian, modifikasi struktural secara presisi dilakukan pada lapisan *linear* yang berada di ujung modul klasifikasi. Sistem membuang konfigurasi 1.000 kelas keluaran bawaan dan mensubstitusinya dengan lapisan *linear* baru. Lapisan modifikasi ini dikonfigurasi untuk tetap menerima 1.280 fitur masukan dari lapisan sebelumnya, namun secara matematis mereduksinya menjadi hanya 4 fitur keluaran (`out_features`). Keempat titik keluaran ini dibangun untuk secara spesifik merepresentasikan distribusi probabilitas dari kelas target penelitian, yakni *female fake*, *female real*, *male fake*, dan *male real*.

### 5.3.2. Parameter Pelatihan

Setelah arsitektur model berhasil dimodifikasi pada bagian lapisan klasifikasinya, tahapan esensial selanjutnya adalah melakukan penentuan status pembaruan bobot jaringan. Berdasarkan hasil komputasi struktural dari sistem, arsitektur EfficientNet-B0 yang telah disesuaikan tersebut memiliki total 4.012.672 parameter matematis. Keseluruhan parameter ini merepresentasikan himpunan bobot (*weights*) dan bias komputasional yang terdistribusi secara hierarkis, mulai dari blok konvolusi awal yang mengekstraksi fitur dasar hingga lapisan *fully connected* di bagian akhir yang bertugas mengambil keputusan klasifikasi.

Dalam implementasi algoritma *transfer learning*, terdapat dua strategi utama terkait penanganan parameter praleatih (*pre-trained parameters*), yakni pembekuan sebagian lapisan jaringan (*layer freezing*) atau penyesuaian bobot secara

menyeluruh (*full fine-tuning*). Pada penelitian ini, sistem dikonfigurasi untuk mengeksekusi strategi *full fine-tuning*, di mana seluruh parameter yang berjumlah 4.012.672 tersebut dipertahankan dalam status aktif dan dapat dilatih (*trainable*). Secara terprogram, hal ini diimplementasikan dengan memastikan bahwa atribut komputasi gradien (`requires_grad`) pada seluruh parameter matriks bernilai benar (`True`), sehingga tidak ada satu pun lapisan jaringan yang dibekukan selama proses pelatihan iteratif berlangsung.

Pemilihan strategi penyesuaian bobot secara menyeluruh ini dilandasi oleh kompleksitas visual dari tugas klasifikasi citra *deepfake*. Meskipun bobot dasar dari ImageNet sudah sangat optimal dalam mengenali bentuk makro benda secara umum, pendeteksian manipulasi wajah buatan AI menuntut kapabilitas analisis pada tingkat tekstur mikroskopis seperti inkonsistensi pori-pori kulit, anomali pencahayaan fokal, dan artefak piksel yang sangat subtil. Dengan menjadikan seluruh lapisan jaringan bersifat *trainable*, model tidak hanya menyesuaikan lapisan akhirnya saja, melainkan juga secara aktif memaksa lapisan konvolusi dasar untuk beradaptasi, mengkalibrasi ulang *filter* spasialnya, dan mempelajari representasi fitur yang sangat spesifik terhadap karakteristik wajah asli maupun *deepfake* pada profil masyarakat Indonesia.

### 5.3.3. *Forward Pass* dan Fungsi Aktivasi

Untuk memvalidasi integritas struktural dan komputasional dari arsitektur yang telah dimodifikasi, sistem menjalankan prosedur simulasi aliran maju (*forward pass validation*) sebelum memasuki fase pelatihan iteratif. Langkah pengujian awal ini bersifat krusial untuk memastikan bahwa tidak ada anomali dimensi matriks (*dimension mismatch*) saat data mengalir dari lapisan konvolusi awal hingga ke lapisan keluaran baru. Pada tahap ini, model untuk sementara waktu dialihkan ke mode evaluasi dan fungsi pelacakan gradien matematis dinonaktifkan (`torch.no_grad()`). Pendekatan ini dilakukan untuk menghemat alokasi memori komputasi pada kartu grafis (GPU) karena pembaruan bobot memang belum diperlukan pada tahap validasi arsitektur.

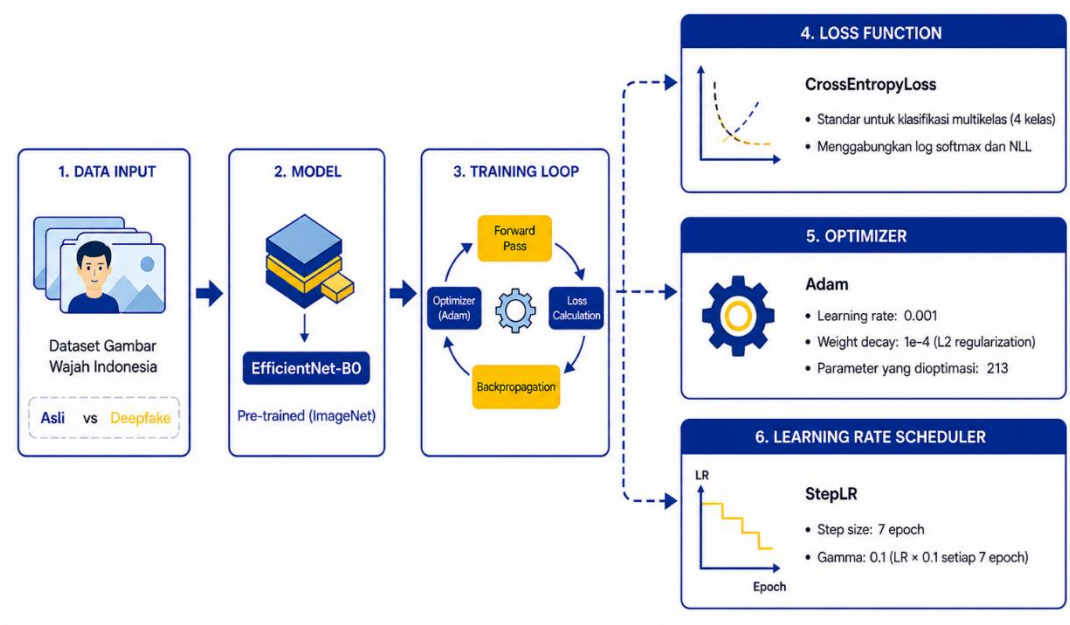
Dalam eksekusi operasionalnya, sistem mengumpankan satu tumpukan data uji coba (*batch*) yang diekstraksi dari modul pemuat data latih. Tumpukan masukan tersebut direpresentasikan dalam bentuk tensor multidimensi [32, 3, 224, 224], yang

mengindikasikan kehadiran 32 sampel citra beresolusi 224 x 224 piksel dengan tiga saluran warna standar (RGB). Ketika tensor ini diproses melintasi seluruh lapisan jaringan EfficientNet-B0, model terbukti mampu memetakan fitur visual dengan sukses dan menghasilkan keluaran berupa nilai logaritma natural mentah (*raw logits*) dengan dimensi matriks [32, 4]. Dimensi keluaran ini memverifikasi bahwa model telah secara presisi memproyeksikan setiap gambar dari total 32 masukan tersebut ke dalam empat nilai numerik yang berkorespondensi langsung dengan empat kelas target penelitian.

Dikarenakan nilai logits yang diekstraksi dari lapisan *linear* masih berupa angka riil tanpa batas rentang yang terstandardisasi, sistem mengimplementasikan fungsi aktivasi *softmax* pada lapisan keluaran tersebut. Secara matematis, operasi eksponensial dalam algoritma *softmax* bertugas mengonversi nilai logits mentah menjadi distribusi probabilitas yang dinormalisasi pada rentang 0 hingga 1 (atau 0% hingga 100%), di mana akumulasi probabilitas untuk keseluruhan kelas komputasi akan selalu bernilai tepat 100%. Pada tahap inferensi final, sistem menerapkan fungsi pencarian nilai maksimum secara *linear* terhadap matriks probabilitas tersebut, dan kelas dengan persentase kalkulasi tertinggi akan ditetapkan secara definitif sebagai hasil prediksi klasifikasi wajah dari arsitektur sistem.

### 5.4. *Loss Function* dan *Optimizer*

Tahap pelatihan (*training*) merupakan fase paling esensial dalam alur kerja pemelajaran mesin, di mana arsitektur EfficientNet-B0 secara iteratif mengkalibrasi bobot parameter komputasionalnya untuk membedakan karakteristik visual antara citra wajah asli dan *deepfake*. Dalam penelitian ini, skenario pelatihan dirancang secara terstruktur melalui dua proses utama yang saling berkesinambungan. Proses pertama difokuskan pada tahap inisialisasi matematis, yang mencakup kegiatan mendefinisikan fungsi kerugian (*loss function*), algoritma optimasi (*optimizer*), dan penjadwalan laju pembelajaran (*learning rate scheduler*). Ketiga komponen fundamental ini dikonfigurasi di awal untuk menjadi kompas panduan bagi arsitektur jaringan, memastikan bahwa arah pembaruan parameter dapat berlangsung secara stabil, efisien, dan memiliki ketahanan terhadap risiko model menghafal data (*overfitting*).




**Gambar 5.7.** *Loss Function*, *Optimizer*, dan *Learning Rate*

### 5.4.1. *Loss Function*

Penelitian ini menggunakan fungsi kerugian *Cross Entropy* (`CrossEntropyLoss`) sebagai standar metrik untuk penyelesaian tugas klasifikasi empat kelas target. Fungsi ini dipilih karena mampu menggabungkan dua operasi perhitungan secara langsung di dalam sistem PyTorch, yakni fungsi *Log Softmax* dan *Negative Log Likelihood* (NLL). Dengan cara ini, model tidak perlu lagi menambahkan fungsi *Softmax* secara manual pada lapisan akhir. Fungsi `CrossEntropyLoss` dirancang untuk langsung memproses nilai prediksi mentah (*raw logits*) menjadi nilai probabilitas yang stabil, untuk kemudian diukur tingkat penyimpangannya terhadap label data yang sebenarnya (*ground truth*).

Secara operasional, fungsi kerugian ini memberikan hukuman (penalti) yang terukur terhadap setiap kesalahan prediksi, khususnya ketika model menghasilkan tebakan yang salah namun dengan tingkat keyakinan yang sangat tinggi (*overconfident*). Semakin jauh tebakan model dari label aslinya, maka nilai kerugian (*loss*) yang terkumpul akan semakin besar. Nilai kerugian yang tinggi ini akan memicu pembaruan bobot parameter yang lebih besar pada saat fase perambatan balik (*backpropagation*). Mekanisme koreksi ini secara dinamis memaksa arsitektur jaringan untuk terus mengkalibrasi ulang representasi fiturnya pada setiap iterasi (*epoch*), hingga probabilitas keluaran sistem semakin konvergen dan akurat dalam membedakan detail visual antara citra wajah asli dan *deepfake*.

### 5.4.2. *Optimizer*

Pembaruan bobot jaringan selama fase perambatan balik (*backward pass*) dijalankan menggunakan algoritma optimasi Adam (*Adaptive Moment Estimation*). Algoritma ini dipilih karena memiliki keunggulan dalam menyesuaikan kecepatan belajar secara otomatis dan terpisah untuk setiap kelompok parameter. Dalam arsitektur yang digunakan pada penelitian ini, terdapat 213 kelompok parameter aktif yang nilainya terus diperbarui. Berbeda dengan metode optimasi standar yang menggunakan satu nilai kecepatan tetap untuk seluruh jaringan, Adam mampu mempercepat atau memperlambat pembaruan pada bobot tertentu berdasarkan riwayat perubahannya di putaran sebelumnya sehingga proses pencarian tingkat kesalahan terendah dapat berjalan jauh lebih cepat dan stabil.

Sebagai langkah awal, nilai laju pembelajaran (*initial learning rate*) pada algoritma optimasi ini ditetapkan pada angka 0.001. Angka ini merupakan standar empiris yang banyak direkomendasikan karena mampu memberikan keseimbangan pergerakan yang sangat baik pada awal masa pelatihan. Nilai 0.001 memastikan bahwa langkah pembaruan bobot tidak terlalu besar maupun terlalu kecil. Jika langkah pembaruan terlalu besar, perhitungan matematis dapat menjadi tidak stabil dan model bisa gagal menemukan titik akurasi terbaik. Sebaliknya, jika nilainya terlalu kecil, proses pelatihan akan memakan waktu yang sangat lama dan rentan berhenti sebelum model selesai belajar secara maksimal.

Selain mengatur kecepatan belajar, algoritma optimasi ini juga dilengkapi dengan mekanisme pencegahan fenomena menghafal data (*overfitting*) melalui parameter penurunan bobot (*weight decay*) yang bernilai $1 \times 10^{-4}$. Parameter ini berfungsi sebagai teknik regularisasi matematis (regularisasi $L2$) yang bertugas mengontrol pertumbuhan nilai bobot di dalam model. Secara prinsip kerja, teknik ini akan memberikan sanksi berupa nilai pengurangan pada matriks bobot yang tumbuh menjadi terlalu besar secara tidak wajar. Pembatasan ini sangat penting untuk menjaga struktur model tetap sederhana. Dengan demikian, model akan benar-benar mempelajari pola umum dari wajah asli dan *deepfake*, bukan sekadar menghafal rincian piksel spesifik dari gambar yang ada di dalam data latih.

### 5.4.3. *Learning Rate Scheduler*

Pengaturan kecepatan belajar (*learning rate*) yang tetap dari awal hingga akhir pelatihan sering kali menimbulkan kendala ketika model sudah mulai mendekati titik akurasi terbaiknya. Jika langkah pembaruan bobot tetap dibiarkan besar pada tahap akhir pelatihan, perhitungan model berisiko melompat melewati titik sasaran yang optimal. Kondisi ini akan menyebabkan nilai kesalahan (*loss*) bergerak naik turun tanpa arah atau terjebak pada angka tertentu (*plateau*). Untuk mengatasi hal tersebut, penelitian ini menerapkan penjadwalan laju pembelajaran yang dapat berubah secara dinamis menggunakan metode `StepLR`. Metode ini memastikan bahwa model dapat memodifikasi bobotnya dengan cepat di awal pembelajaran, namun menjadi jauh lebih berhati-hati dan presisi seiring bertambahnya waktu pelatihan.

Secara teknis, metode `StepLR` bekerja dengan cara memotong nilai kecepatan belajar secara berkala berdasarkan rentang waktu yang telah ditentukan sebelumnya. Pada pengaturan program yang digunakan, sistem diberikan parameter jeda langkah (*step size*) sebanyak 7 putaran dan faktor pengali (*gamma*) sebesar 0.1. Kombinasi kedua parameter ini memberikan instruksi kepada algoritma optimasi untuk memantau jalannya proses pelatihan, lalu mengalikan nilai laju pembelajaran yang sedang aktif dengan 0.1 setiap kali pelatihan telah melewati kelipatan tujuh putaran (*epoch*). Penurunan nilai secara otomatis ini menggantikan kebutuhan penyesuaian manual yang sering kali tidak efisien.

Sebagai gambaran penerapan nyata dari pengaturan tersebut, pada putaran ke-0 hingga ke-6, model akan menggunakan kecepatan belajar awal yang cukup besar yaitu 0.001. Nilai awal ini memungkinkan model untuk dengan cepat mengenali pola-pola dasar yang membedakan wajah asli dan *deepfake*. Begitu memasuki putaran ke-7 hingga ke-13, sistem akan secara otomatis memperkecil kecepatan tersebut menjadi 0.0001, dan akan terus mengecil sepuluh kali lipat pada setiap kelipatan tujuh putaran berikutnya. Penurunan langkah secara bertahap ini sangat penting untuk membantu model melakukan penyesuaian parameter yang jauh lebih halus (*fine-tuning*). Dengan demikian, pencapaian tingkat kesalahan terendah dapat berlangsung dengan sangat stabil dan hasil prediksi akhir menjadi lebih akurat.

### 5.4.4. *Hyperparameters*

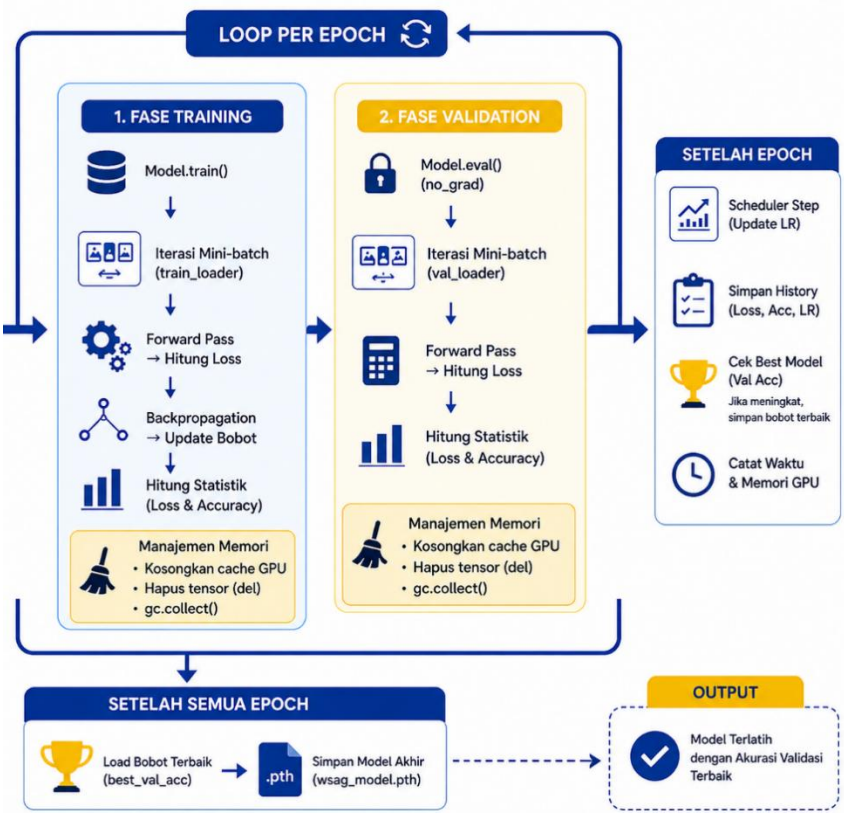
Secara keseluruhan, parameter pelatihan diatur untuk menjalankan proses komputasi selama 15 putaran penuh atau yang sering disebut sebagai *epoch*. Satu putaran penuh berarti model telah melihat dan mempelajari seluruh gambar yang ada di dalam data latih tepat satu kali. Pemilihan angka 15 ini merupakan batas yang dinilai memadai untuk memberikan waktu bagi arsitektur EfficientNet-B0 dalam mengenali pola perbedaan antara wajah asli dan *deepfake* secara mendalam. Jika jumlah putaran dibuat terlalu sedikit, model mungkin belum selesai belajar sehingga tingkat akurasinya masih rendah (*underfitting*). Sebaliknya, jika putaran dibuat terlalu banyak, model berisiko sekadar menghafal gambar data latih dan justru kehilangan kemampuannya untuk mengenali gambar baru (*overfitting*).

Selama proses pembelajaran berlangsung, seluruh data latih tidak dimasukkan sekaligus secara bersamaan ke dalam model demi menjaga kestabilan memori komputer. Sebanyak 720 gambar pada data latih dibagi ke dalam beberapa kelompok yang lebih kecil menggunakan ukuran tumpukan (*batch size*) sebanyak 32 gambar per proses perhitungan. Pembagian ini secara otomatis menghasilkan 23 langkah komputasi (*steps*) di setiap satu putaran penuh. Dengan rancangan 15 putaran dan 23 langkah per putarannya, program secara keseluruhan akan menjalankan 345 tahapan optimasi hingga seluruh fase pelatihan dinyatakan selesai. Pengaturan parameter ini dirancang secara khusus untuk menyeimbangkan durasi waktu pelatihan sekaligus memastikan pembaruan bobot model dapat berjalan secara teratur tanpa membebani kapasitas memori perangkat keras secara berlebihan.

## 5.5. Pelatihan Model

Setelah mendefinisikan *loss function*, *optimizer*, dan *learning rate scheduler*, selanjutnya adalah eksekusi iterasi pelatihan atau yang lebih dikenal dengan istilah *loop training*. Pada tahapan dinamis ini, matriks citra dari aliran data latih dan data validasi dimasukkan ke dalam model secara bergantian pada setiap siklus putaran (*epoch*). Mekanisme *loop training* mencakup serangkaian operasi matematis berkelanjutan, mulai dari perambatan maju (*forward pass*) untuk menghasilkan luaran prediksi, kalkulasi nilai kesalahan (*loss computation*), hingga perambatan balik (*backward pass*) guna memperbarui bobot jaringan. Secara simultan, kinerja

model juga dievaluasi menggunakan data validasi pada akhir setiap siklus untuk memonitor tren peningkatan akurasi secara objektif. Melalui integrasi proses ini, model diharapkan mampu mencapai titik konvergensi yang optimal.




**Gambar 5.8.** Proses *Loop Training*

Proses utama pelatihan dijalankan secara berulang selama 15 putaran (*epochs*). Pada setiap putarannya, sistem menjalankan dua tahapan secara bergantian, yaitu fase pelatihan dan fase validasi. Pada fase pelatihan, model diatur ke dalam mode belajar. Komputer secara bertahap mengambil tumpukan gambar dari aliran data latih, membuat prediksi keluaran, dan menghitung seberapa besar tingkat kesalahannya. Dari nilai kesalahan tersebut, program melakukan perambatan balik (*backward pass*) untuk memperbarui nilai bobot jaringan agar prediksi model selanjutnya menjadi lebih tepat. Selama proses ini berlangsung, pembersihan memori sementara (*cache*) pada GPU dilakukan secara berkala agar kapasitas memori komputer tetap stabil dan mencegah terjadinya penumpukan beban komputasi.

Setelah satu putaran pelatihan selesai, sistem langsung beralih ke fase validasi. Pada tahapan ini, model diatur ke dalam mode evaluasi untuk diuji kemampuannya menggunakan kumpulan data validasi. Berbeda dengan fase

pelatihan, pada fase validasi sistem sama sekali tidak melakukan pembaruan bobot pembelajaran. Oleh karena itu, proses perhitungan gradien matematis dinonaktifkan secara eksplisit menggunakan perintah `torch.no_grad()`. Penonaktifan ini sangat krusial untuk mencegah sistem membangun grafik perhitungan (*computation graph*) yang merekam riwayat kalkulasi. Jika perintah ini dilewatkan, memori GPU akan dialokasikan secara sia-sia untuk menyimpan perhitungan yang tidak digunakan, yang berpotensi menyebabkan kehabisan memori (*Out of Memory*) dan memperlambat proses pengujian. Melalui optimalisasi ini, proses evaluasi akurasi dapat berjalan dengan sangat cepat dan efisien.

Sebagai langkah terakhir di setiap akhir putaran, program akan membandingkan hasil akurasi validasi terbaru dengan rekor akurasi sebelumnya. Jika model berhasil mencapai tingkat akurasi pengenalan yang lebih tinggi, maka susunan matriks bobot model pada putaran tersebut akan disalin dan disimpan sebagai rekam jejak model terbaik. Berdasarkan hasil pelaksanaan komputasi, akurasi tertinggi tercapai pada putaran keempat dengan persentase keberhasilan 100%. Susunan bobot terbaik inilah yang kemudian secara otomatis dikembalikan ke dalam model dan disimpan secara permanen ke dalam ruang penyimpanan virtual dengan nama berkas `wsag_model.pth`.

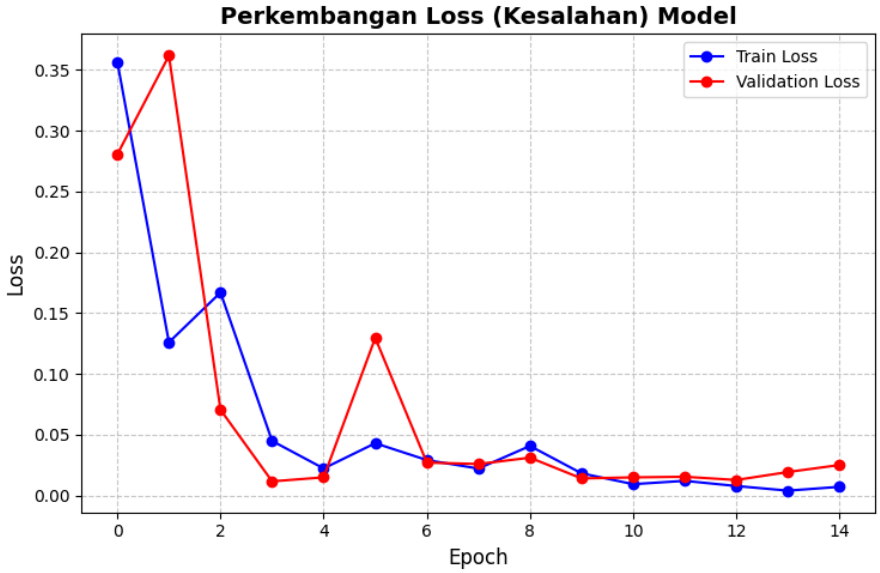
## 5.6. Hasil Pelatihan Model

Bagian ini menyajikan analisis terhadap kinerja model EfficientNet-B0 selama fase pelatihan yang berlangsung sebanyak 15 *epoch*. Evaluasi performa model dilakukan dengan memantau pergerakan nilai kerugian (*loss*) dan tingkat akurasi (*accuracy*) pada dua kumpulan data, yaitu data latih (*training*) dan data validasi (*validation*). Visualisasi dari hasil pelatihan tersebut disajikan pada Gambar 5.9 dan Gambar 5.10.

### 5.6.1. Analisis Kurva Nilai Kerugian (*Loss*)

Berdasarkan grafik perkembangan *loss* (kesalahan) model pada Gambar 5.9, analisis terhadap pergerakan nilai kerugian pada fase pelatihan (*train loss*) menunjukkan performa adaptasi model yang sangat impresif. Pada awal proses komputasi (*epoch* ke-0 hingga ke-1), terlihat penurunan nilai kesalahan yang sangat tajam dari angka 0.35 menjadi sekitar 0.12. Penurunan drastis ini merupakan indikator positif yang menunjukkan bahwa algoritma optimasi Adam merespons

dengan cepat dalam menyesuaikan bobot awal jaringan untuk mengenali pola-pola dasar visual yang membedakan antara citra wajah asli dan *deepfake*. Seiring bertambahnya putaran iterasi, tren penurunan ini terus berlanjut secara konsisten dan perlahan melandai hingga mencapai titik kesalahan yang nyaris menyentuh angka nol pada akhir fase pelatihan. Pergerakan yang stabil pada fase akhir ini membuktikan bahwa proses pembaruan parameter berjalan secara presisi tanpa mengalami lompatan yang tidak terarah.



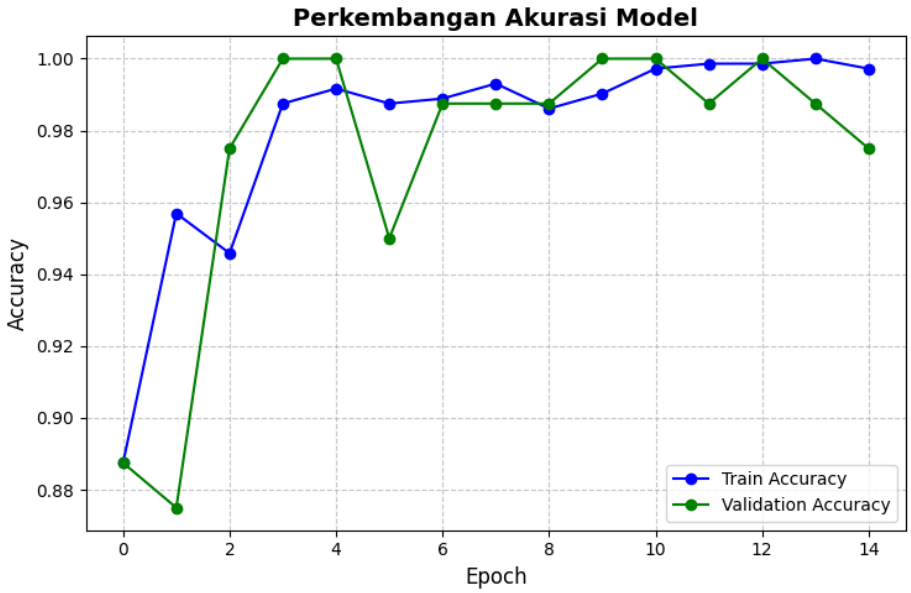

**Gambar 5.9.** Grafik Perkembangan *Loss*

Berbeda dengan kurva pelatihan yang bergerak sangat mulus, pergerakan kurva nilai kerugian pada data validasi (*validation loss*) menampilkan dinamika yang sedikit lebih fluktuatif pada paruh pertama masa pelatihan. Lonjakan nilai kesalahan yang cukup signifikan teramati secara khusus pada *epoch* ke-1 dan kembali berulang pada *epoch* ke-5. Fluktuasi matematis semacam ini merupakan fenomena wajar dalam skenario pelatihan model pembelajaran mendalam (*deep learning*). Kondisi tersebut umumnya terjadi ketika model sedang berusaha melakukan adaptasi terhadap variasi fitur-fitur gambar baru yang ada di dalam data validasi atau sebagai bentuk respons terhadap penyesuaian laju pembelajaran yang masih cukup besar di awal iterasi. Meskipun sempat mengalami lonjakan sesaat, arsitektur jaringan terbukti mampu mengoreksi dirinya sendiri dengan sangat cepat. Hal ini terlihat dari grafik kurva yang langsung kembali menurun tajam dan terkendali pada putaran-putaran berikutnya.

Memasuki paruh kedua pelatihan, khususnya setelah *epoch* ke-7 hingga putaran yang terakhir, kurva *validation loss* tampak semakin melandai dan mulai bergerak sangat beriringan dengan kurva *train loss*. Jarak simpangan (*gap*) yang sangat sempit dan konsisten antara kedua kurva pada akhir iterasi ini memberikan pembuktian krusial mengenai kualitas sistem yang telah dibangun. Kestabilan pada titik akhir ini mengonfirmasi bahwa arsitektur jaringan tidak sekadar terjebak dalam fase menghafal rincian piksel data latih, melainkan benar-benar berhasil menggeneralisasi fitur-fitur esensial secara utuh. Konvergensi yang harmonis antara nilai kerugian pelatihan dan validasi ini sekaligus menjadi bukti matematis bahwa model EfficientNet-B0 telah menguasai keseimbangan pemelajaran secara optimal dan tidak mengalami degradasi performa.

### 5.6.2. Analisis Kurva Akurasi

Berdasarkan grafik perkembangan akurasi model pada Gambar 5.10, evaluasi terhadap tingkat akurasi pelatihan (*train accuracy*) memperlihatkan pencapaian yang sangat memuaskan sejak awal proses komputasi. Pada putaran pertama (*epoch* ke-0), model telah mampu mencatatkan tingkat akurasi awal yang tergolong tinggi yaitu di sekitar angka 88.7%. Tingginya akurasi awal ini merupakan indikasi positif dari penggunaan arsitektur EfficientNet-B0 yang memanfaatkan metode pembelajaran transfer (*transfer learning*), di mana model tidak memulai proses pembelajarannya dari nol melainkan menggunakan bekal pengetahuan visual dari pratamalatih ImageNet. Seiring dengan berjalannya proses iterasi, kurva akurasi pelatihan terus merangkak naik secara stabil dan konsisten hingga berhasil menyentuh angka di atas 99% pada akhir fase pelatihan. Peningkatan yang persisten ini mengonfirmasi bahwa jaringan saraf tiruan secara efektif mampu mengekstraksi dan mempelajari berbagai fitur kompleks yang membedakan kelas wajah asli dan *deepfake* secara komprehensif.




**Gambar 5.10.** Grafik Perkembangan Akurasi

Performa yang tinggi pada data latih tersebut juga selaras dengan kinerja model saat dihadapkan pada kumpulan data validasi. Kurva *validation accuracy* (garis hijau) menampilkan pergerakan adaptif yang sangat baik pada paruh pertama pelatihan. Setelah sedikit penurunan wajar pada *epoch* ke-1 yang merupakan fase penyesuaian bobot awal, akurasi model langsung melonjak tajam dan berhasil menyentuh tingkat keberhasilan sempurna yaitu 100% pada beberapa titik iterasi seperti pada *epoch* ke-3, ke-4, dan ke-9. Pencapaian akurasi pada fase validasi ini merupakan temuan yang sangat penting. Hal ini membuktikan bahwa kemampuan model dalam mengenali pola rekayasa visual tidak terbatas pada gambar-gambar yang telah dipelajarinya berulang kali, melainkan juga sangat tangguh dan adaptif ketika digunakan untuk menebak data baru yang sama sekali belum pernah dilihat sebelumnya.

Memasuki putaran-putaran akhir komputasi, kurva akurasi validasi masih terus bergerak stabil di atas ambang batas 95% dengan posisi akhir berada di sekitar angka 97.5%. Meskipun terdapat sedikit fluktuasi atau penurunan minor pada *epoch* terakhir jika dibandingkan dengan titik puncak akurasinya, pergerakan ini masih murni berada di dalam batas toleransi kewajaran komputasi (*computational variance*) dan tidak mengurangi kualitas arsitektur secara keseluruhan. Jarak simpangan yang sangat rapat antara kurva akurasi pelatihan dan validasi di sepanjang rentang iterasi ini memberikan validasi empiris bahwa arsitektur model

tidak terjebak dalam masalah penghafalan data. Sebaliknya, model terbukti sukses dalam membentuk batas keputusan klasifikasi (*decision boundary*) yang sangat kuat, presisi, dan dapat diandalkan secara ilmiah sebagai sistem pendeteksi manipulasi wajah.

### 5.6.3. Analisis Kondisi Model (*Fitting*)

Berdasarkan analisis terhadap pergerakan metrik evaluasi pada fase pelatihan dan validasi, arsitektur model menunjukkan kondisi pembelajaran yang baik (*good fit*). Selama proses pelatihan, tidak ditemukan indikasi *underfitting*, yaitu kondisi ketika model gagal mempelajari pola data secara memadai yang umumnya ditandai oleh nilai *loss* tinggi dan akurasi yang rendah. Model EfficientNet-B0 menunjukkan kemampuan dalam mempelajari representasi fitur dari data latih, yang tercermin dari peningkatan akurasi selama pelatihan serta penurunan nilai *loss* secara konsisten. Hasil ini mengindikasikan bahwa kapasitas arsitektur yang digunakan memadai untuk menangani kompleksitas data pada penelitian ini.

Selain tidak menunjukkan indikasi *underfitting*, hasil evaluasi juga mengindikasikan bahwa model tidak mengalami *overfitting*. Dalam konteks *deep learning*, *overfitting* terjadi ketika model terlalu menyesuaikan diri terhadap data latih sehingga kinerjanya menurun pada data baru. Kondisi ini umumnya ditandai dengan penurunan *train loss* yang terus berlanjut, sementara *validation loss* meningkat secara signifikan. Pada penelitian ini, kurva *train loss* dan *validation loss* menunjukkan pola yang relatif konvergen hingga akhir pelatihan. Nilai *validation loss* tetap stabil pada tingkat yang rendah dan tidak menunjukkan peningkatan yang tajam. Selisih yang kecil antara metrik pelatihan dan validasi mengindikasikan bahwa model mampu mempelajari pola fitur yang relevan tanpa bergantung secara berlebihan pada karakteristik spesifik data latih.

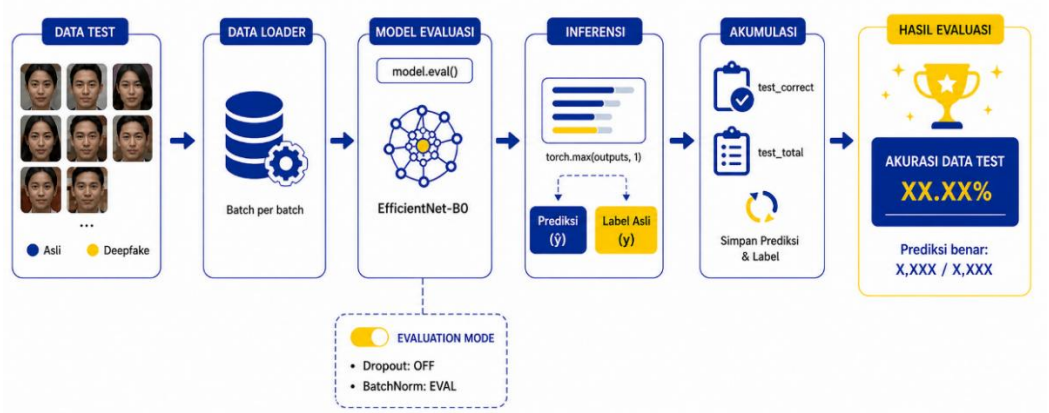
Pencapaian kondisi *good fit* mengindikasikan bahwa model memiliki kemampuan generalisasi yang baik. Kemampuan ini menjadi salah satu aspek penting dalam pengembangan sistem klasifikasi, karena berkaitan dengan kemampuan model dalam mempertahankan kinerja pada data baru di luar data latih. Stabilitas akurasi validasi yang tinggi serta capaian akurasi hingga 100% pada beberapa iterasi menunjukkan bahwa model mampu mempelajari karakteristik fitur

yang relevan dalam membedakan wajah asli dan *deepfake*, termasuk pola tekstur dan pencahayaan yang berperan dalam pembentukan *decision boundary*.

## 5.7. Evaluasi Kinerja Model

Setelah melalui fase pelatihan dengan hasil konvergensi yang optimal, kinerja arsitektur EfficientNet-B0 dievaluasi lebih lanjut menggunakan data uji (*test set*) yang belum pernah dilihat oleh model sebelumnya. Pengujian ini berperan penting dalam menilai kemampuan generalisasi model saat dihadapkan pada data baru. Sebelum aliran data uji diproses, model terlebih dahulu diatur ke dalam mode evaluasi secara terprogram. Pengaturan ini sangat penting karena akan menonaktifkan fungsi *dropout*, sehingga seluruh koneksi neuron di dalam jaringan aktif sepenuhnya untuk memberikan prediksi yang paling maksimal. Selain itu, sistem juga menghentikan secara eksplisit fungsi pelacakan gradien matematis. Karena model tidak lagi membutuhkan pembaruan parameter, penonaktifan gradien ini akan sangat menghemat alokasi memori kartu grafis dan mempercepat waktu komputasi pengujian.

Pada tahap operasionalnya, sebanyak 200 gambar dari direktori data uji dimasukkan ke dalam model secara bertahap. Sistem kemudian mengambil tebakan prediksi dari lapisan akhir model dan membandingkannya secara langsung dengan label kebenaran dasar (*ground truth*) dari setiap gambar. Selama proses ini berlangsung, seluruh riwayat tebakan model dan label asli disimpan ke dalam ruang penyimpanan sementara untuk kebutuhan analisis metrik yang lebih mendalam pada tahapan berikutnya.




**Gambar 5.11.** Proses Evaluasi Model pada Data Uji

Untuk memperoleh evaluasi yang komprehensif dan objektif, kinerja model dianalisis menggunakan tiga metrik utama yaitu *confusion matrix* untuk memetakan

hasil prediksi, *classification report* untuk analisis statistik secara rinci, serta kurva *Receiver Operating Characteristic* (ROC) beserta nilai *Area Under the Curve* (AUC) untuk mengukur kinerja klasifikasi secara keseluruhan.

### 5.7.1. *Confusion Matrix*

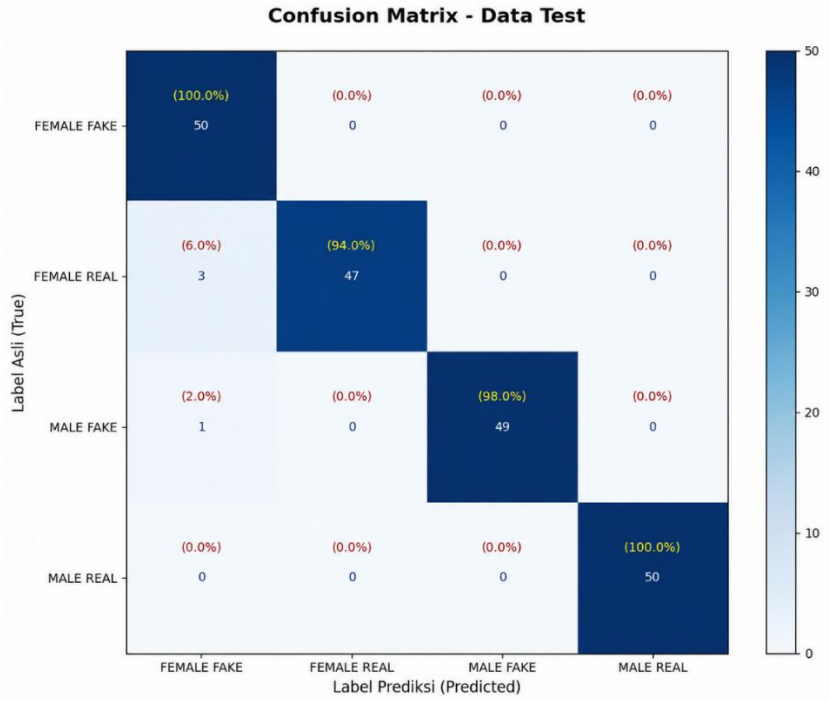
Setelah evaluasi model pada data uji, selanjutnya adalah melakukan pemetaan hasil prediksi menggunakan matriks kebingungan (*confusion matrix*). *Confusion matrix* merupakan alat evaluasi visual yang digunakan untuk membandingkan label kelas asli (*ground truth*) dengan label prediksi model. Melalui matriks ini, kinerja model tidak hanya dinilai dari jumlah prediksi yang benar, tetapi juga dari kemampuannya dalam mengklasifikasikan setiap kelas secara spesifik serta mengidentifikasi pola kesalahan (*misclassification*) yang terjadi.

Secara teknis, pemetaan ini dilakukan menggunakan fungsi dari pustaka `scikit-learn` yang membandingkan secara langsung daftar label asli dari data uji dengan daftar label hasil tebakan model. Karena dataset penelitian ini terbagi menjadi empat kategori yaitu *female fake*, *female real*, *male fake*, dan *male real*, matriks yang dihasilkan secara otomatis memiliki dimensi 4x4. Pada struktur matriks ini, sumbu mendatar (kolom) mewakili label tebakan yang dihasilkan oleh model, sedangkan sumbu vertikal (baris) mewakili label asli dari data uji.

Untuk memudahkan proses analisis, susunan angka pada matriks tersebut divisualisasikan menjadi bentuk grafik *heatmap* menggunakan pustaka komputasi `matplotlib`. Visualisasi ini menggunakan skema gradasi warna biru, di mana intensitas warna yang semakin gelap menunjukkan jumlah penumpukan data yang semakin banyak pada titik tersebut. Sel-sel yang berada pada garis diagonal utama dari kiri atas ke kanan bawah merupakan area yang menampilkan jumlah tebakan model yang benar. Sebaliknya, sel-sel yang berada di luar garis diagonal utama menunjukkan letak kesalahan klasifikasi, sehingga peneliti dapat melihat secara pasti ke kelas mana model sering kali keliru dalam memberikan tebakan.

Untuk memberikan pemahaman yang lebih komprehensif, grafik visualisasi ini juga dilengkapi dengan keterangan persentase perhitungan pada setiap selnya. Nilai persentase ini diperoleh dengan cara membagi jumlah prediksi pada suatu sel dengan total keseluruhan gambar yang ada pada baris kelas aslinya. Selain ditampilkan melalui grafik, sistem juga diprogram untuk mencetak rincian jumlah

gambar yang berhasil ditebak dengan benar beserta persentasenya untuk masing-masing kelas secara tekstual. Penjabaran rincian ini memiliki peran yang sangat penting untuk memastikan apakah model memiliki tingkat kecerdasan yang seimbang pada keempat kelas target, ataukah terdapat ketimpangan di mana kelas tertentu terbukti lebih sulit untuk diklasifikasikan dibandingkan kelas lainnya. Hasil pemetaan prediksi model pada 200 data uji disajikan pada Gambar 5.12.




**Gambar 5.12.** Visualisasi *Confusion Matrix* pada Data Uji

Berdasarkan visualisasi pada Gambar 5.12, matriks berukuran 4×4 merepresentasikan empat kelas target, yaitu *female fake*, *female real*, *male fake*, dan *male real*. Sumbu vertikal menunjukkan label asli citra, sedangkan sumbu horizontal menunjukkan label prediksi yang dihasilkan oleh model. Sel-sel pada diagonal utama (dari kiri atas ke kanan bawah) yang berwarna biru gelap merepresentasikan jumlah citra yang berhasil diklasifikasikan dengan benar. Secara keseluruhan, model menghasilkan 196 prediksi benar dari 200 citra uji, dengan tingkat akurasi sebesar 98.0%.

Rincian tingkat keberhasilan pada masing-masing kelas menunjukkan variasi kinerja model dalam melakukan klasifikasi. Pada kelas *female fake*, model mampu mengklasifikasikan seluruh citra dengan benar, yaitu 50 dari 50 citra (akurasi 100%). Pada kelas *female real*, model berhasil mengenali 47 citra dengan benar (94.0%), namun masih terdapat 3 citra (6.0%) yang salah diklasifikasikan sebagai

*female fake*. Selanjutnya, pada kelas *male fake* model mengklasifikasikan 49 citra secara tepat (98.0%), dengan 1 citra (2.0%) yang keliru diprediksi sebagai *female fake*. Sementara itu, pada kelas *male real* model kembali mencapai akurasi 100% dengan mengklasifikasikan seluruh 50 citra secara benar.

Meskipun akurasi keseluruhan tergolong tinggi, analisis terhadap sel di luar diagonal utama memberikan informasi penting mengenai keterbatasan model. Dari total empat kesalahan prediksi, seluruhnya terklasifikasi ke dalam kelas *female fake*. Kesalahan pada tiga citra *female real* yang diprediksi sebagai *female fake* mengindikasikan adanya kemiripan fitur visual pada kelas wanita, sehingga model cenderung sensitif terhadap variasi pencahayaan, tekstur kulit, atau noise tertentu yang menyerupai artefak *deepfake*. Selain itu, satu kesalahan pada citra *male fake* yang diprediksi sebagai *female fake* diduga berkaitan dengan distorsi hasil rekayasa *deepfake* yang menghasilkan karakteristik visual, seperti kehalusan kulit atau struktur wajah, yang tumpang tindih dengan representasi fitur wajah wanita dalam model. Meskipun terdapat kecenderungan kesalahan pada kelas *female fake*, jumlah kesalahan ini relatif kecil, sehingga menunjukkan bahwa model EfficientNet-B0 memiliki kinerja yang baik dalam membedakan manipulasi wajah serta karakteristik gender pada data uji.

### 5.7.2. *Classification Report*

Untuk melengkapi evaluasi kinerja model yang telah disajikan melalui *confusion matrix*, analisis dilanjutkan menggunakan *classification report* guna memperoleh informasi statistik yang lebih rinci. Proses perhitungan dijalankan menggunakan fungsi `classification_report` dari pustaka komputasi `scikit-learn`. Fungsi ini secara otomatis memproses seluruh riwayat label asli dan label tebakan model dari tahapan sebelumnya, lalu mengelompokkan hasil perhitungannya ke dalam empat kategori target secara terpisah. Melalui penerapan parameter `target_names`, hasil keluaran program tidak lagi menampilkan angka kelas bawaan mesin seperti 0 atau 1, melainkan langsung menggunakan nama kelas yang sebenarnya sehingga data menjadi jauh lebih mudah untuk dibaca dan diinterpretasikan.

*Classification report* menghasilkan tiga metrik evaluasi utama untuk setiap kelas, yaitu presisi (*precision*), sensitivitas (*recall*), dan skor F1 (*F1-score*). Presisi

mengukur tingkat ketepatan tebakan positif dari model. Sebagai contoh, dari semua gambar yang ditebak oleh model sebagai *female fake*, nilai presisi menunjukkan berapa persen yang tebakannya benar. Sementara itu, sensitivitas mengukur kemampuan model dalam menemukan seluruh data yang relevan. Artinya, dari seluruh gambar *female fake* yang memang ada di dalam data uji, nilai sensitivitas menunjukkan berapa persen yang berhasil ditemukan dan tidak terlewatkan oleh model. Karena sebuah model komputasi sering kali memiliki nilai presisi yang tinggi namun nilai sensitivitasnya rendah atau sebaliknya, laporan ini juga menyajikan skor F1. Skor ini merupakan nilai rata-rata khusus yang menyeimbangkan perhitungan presisi dan sensitivitas, sehingga memberikan gambaran keandalan yang paling adil untuk masing-masing kelas.

Pada rancangan kodenya, sistem juga diberikan pengaturan khusus berupa parameter `digits=4`. Pengaturan ini memberikan instruksi kepada program untuk mencetak hasil perhitungan statistik hingga empat angka di belakang koma, bukan dua angka seperti pada pengaturan bawaan aslinya. Penambahan tingkat ketelitian desimal ini memiliki peran yang sangat penting dalam penelitian klasifikasi yang memiliki tingkat keberhasilan tinggi. Dalam banyak kasus, perbedaan kinerja antar kelas sangatlah kecil. Jika hasil perhitungan hanya menggunakan dua desimal, persentase keberhasilan dapat dengan mudah dibulatkan ke atas menjadi angka sempurna (1.00), yang pada akhirnya menutupi fakta bahwa model tersebut masih melakukan sedikit kesalahan. Dengan menggunakan format empat angka desimal, pengamatan terhadap metrik evaluasi akhir menjadi jauh lebih teliti dan mencerminkan kondisi model yang sesungguhnya. Rincian *classification report* disajikan pada Gambar 5.13, sedangkan visualisasi dalam bentuk grafik batang (*bar chart*) untuk memudahkan perbandingan antar kelas ditampilkan pada Gambar 5.14.

| KELAS | PRECISION (P) | RECALL (R) | F1-SCORE (F1) | SUPPORT (N) |
|---|---|---|---|---|
| FEMALE FAKE | 0.9259 | 1.0000 | 0.9615 | 50 |
| FEMALE REAL | 1.0000 | 0.9400 | 0.9691 | 50 |
| MALE FAKE | 1.0000 | 0.9800 | 0.9899 | 50 |
| MALE REAL | 1.0000 | 1.0000 | 1.0000 | 50 |
| ACCURACY | | | 0.9800 | 200 |
| MACRO AVG | 0.9815 | 0.9800 | 0.9801 | 200 |
| WEIGHTED AVG | 0.9815 | 0.9800 | 0.9801 | 200 |

**Gambar 5.13.** *Classification Report*

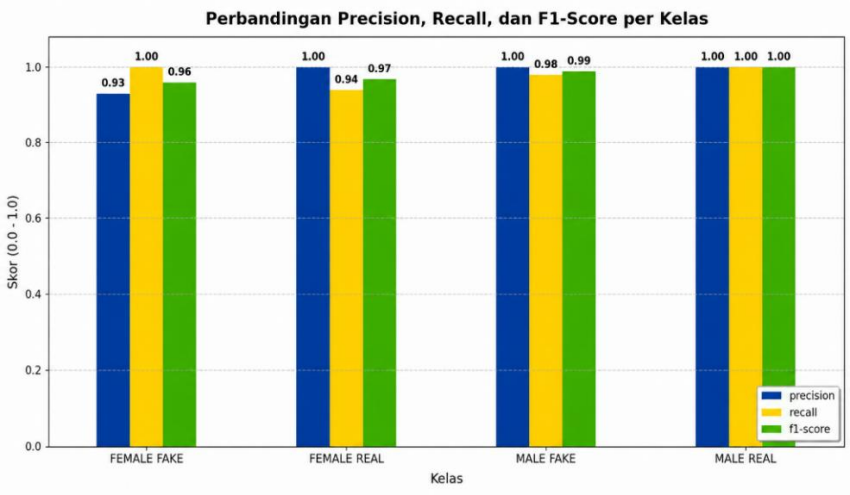
Presisi (*precision*) mengukur tingkat ketepatan prediksi positif yang dihasilkan oleh model, yaitu proporsi prediksi pada suatu kelas yang benar-benar sesuai dengan label aslinya. Berdasarkan Gambar 5.13, model menunjukkan nilai presisi sebesar 1.0000 (100%) pada tiga kelas, yaitu *female real*, *male fake*, dan *male real*, yang menunjukkan bahwa seluruh prediksi pada kelas-kelas tersebut tepat. Sementara itu, pada kelas *female fake* nilai presisi tercatat sebesar 0.9259 (92.59%). Nilai yang lebih rendah ini berkaitan dengan hasil *confusion matrix*, di mana terdapat empat citra dari kelas lain (tiga *female real* dan satu *male fake*) yang keliru diprediksi sebagai *female fake*, sehingga meningkatkan jumlah *false positive* pada kelas tersebut.

Sensitivitas (*recall*) mengukur kemampuan model dalam mengidentifikasi seluruh data positif pada suatu kelas, yaitu proporsi data yang berhasil dikenali dari keseluruhan data yang seharusnya termasuk dalam kelas tersebut. Hasil pengujian menunjukkan bahwa kelas *female fake* dan *male real* memiliki nilai *recall* sebesar 1.0000 (100%), yang berarti seluruh citra pada kedua kelas tersebut berhasil terdeteksi oleh model. Sementara itu, nilai *recall* untuk kelas *female real* sebesar 0.9400 (94.00%) karena terdapat tiga citra yang tidak teridentifikasi dengan benar, dan pada kelas *male fake* sebesar 0.9800 (98.00%) akibat satu citra yang salah diklasifikasikan ke kelas lain (*false negative*).

Skor F1 (*F1-score*) merupakan rata-rata harmonik dari presisi (*precision*) dan sensitivitas (*recall*), sehingga digunakan untuk menyeimbangkan kedua metrik tersebut dalam evaluasi kinerja model. Metrik ini penting karena nilai presisi yang tinggi tidak selalu diikuti oleh nilai *recall* yang tinggi, demikian pula sebaliknya. Berdasarkan Gambar 5.6, seluruh kelas menunjukkan nilai skor F1 yang tinggi yaitu berada di atas 0.96. Kelas *male real* memiliki nilai skor F1 sebesar 1.0000, sedangkan kelas *female fake* mencatat nilai terendah sebesar 0.9615 yang dipengaruhi oleh nilai presisi pada kelas tersebut.

Akurasi mengukur rasio jumlah prediksi yang benar terhadap seluruh data yang diuji. Berdasarkan hasil evaluasi, model EfficientNet-B0 memperoleh akurasi sebesar 0.9800 (98.00%). Nilai ini konsisten dengan hasil pada *confusion matrix*, di mana model berhasil mengklasifikasikan 196 dari 200 citra uji dengan benar. Tingkat akurasi tersebut dapat dianggap representatif dalam menggambarkan kinerja model, mengingat data uji memiliki distribusi kelas yang seimbang dengan masing-masing kelas terdiri dari 50 citra sampel.

Metrik rata-rata makro (*macro average*) dan rata-rata tertimbang (*weighted average*) menunjukkan nilai sebesar 0.9801 (98.01%) untuk seluruh parameter. Konsistensi nilai pada keempat kelas ini mengindikasikan bahwa model EfficientNet-B0 tidak menunjukkan kecenderungan bias akibat ketidakseimbangan kelas (*class imbalance*). Selain itu, hasil tersebut menunjukkan bahwa model memiliki kinerja yang relatif seimbang dalam menganalisis fitur wajah pria dan wanita, serta mampu membedakan antara citra wajah asli dan manipulasi *deepfake* dengan baik.




**Gambar 5.14.** *Per-class Metrics Bar Chart*

Berdasarkan grafik perbandingan *precision*, *recall*, dan *F1-score* per kelas pada Gambar 5.14, terlihat adanya variasi kinerja klasifikasi ketika metrik dianalisis berdasarkan kategori gender. Model EfficientNet-B0 menunjukkan kinerja yang relatif lebih konsisten pada identifikasi citra wajah laki-laki dibandingkan wajah perempuan. Hal ini ditunjukkan oleh kelas *male real* yang memperoleh nilai 1.00 pada seluruh metrik evaluasi. Selain itu, kelas *male fake* juga menunjukkan kinerja yang tinggi dengan nilai *precision* sebesar 1.00, *recall* 0.98, dan *F1-score* 0.99.

Sebaliknya, pada kategori wajah perempuan model menunjukkan variasi kinerja yang lebih terlihat. Kelas *female real* mengalami penurunan nilai *recall* menjadi 0.94, yang menunjukkan bahwa sebagian citra wajah perempuan asli tidak teridentifikasi dengan tepat dan diprediksi sebagai *fake*. Kesalahan ini berdampak pada kelas *female fake*, di mana nilai *precision* menurun menjadi 0.93 akibat adanya prediksi keliru dari kelas *female real*. Meskipun nilai *F1-score* pada kedua kelas perempuan masih berada pada kisaran tinggi (0.96 dan 0.97), perbedaan ini mengindikasikan bahwa model memiliki tingkat kesulitan yang relatif lebih besar dalam menganalisis fitur wajah perempuan dibandingkan wajah laki-laki.

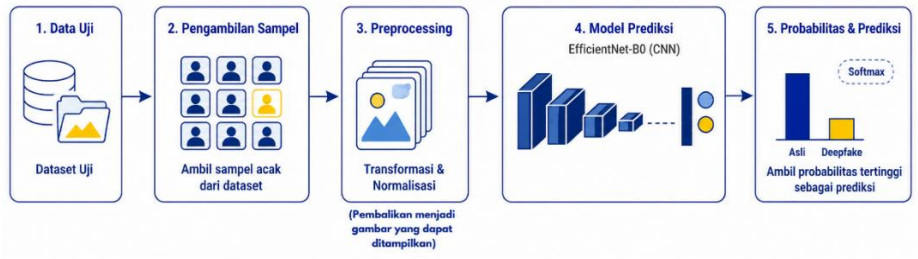
Secara visual, citra wajah perempuan dalam *dataset* cenderung memiliki variasi yang lebih beragam seperti penggunaan riasan (*makeup*), produk perawatan kulit, serta *filter* kamera yang menghasilkan pencahayaan lebih merata dan efek penghalusan tekstur kulit (*skin smoothing*). Karakteristik visual wajah perempuan asli yang cenderung halus dapat memiliki tumpang tindih (*overlap*) fitur dengan artefak yang dihasilkan oleh algoritma *deepfake*. Dalam proses manipulasi wajah, algoritma *deepfake* sering kali menghaluskan transisi piksel sehingga menghasilkan tekstur kulit yang tampak tidak alami. Kemiripan antara efek riasan atau perawatan kulit dengan artefak penghalusan tersebut dapat menyebabkan model mengklasifikasikan sebagian citra wajah perempuan asli sebagai citra manipulasi.

Sebaliknya, citra wajah laki-laki umumnya menampilkan tekstur mikro yang lebih kontras, pori-pori yang lebih jelas, struktur rahang yang tegas, serta keberadaan folikel rambut wajah seperti kumis atau janggut. Karakteristik ini relatif lebih sulit direplikasi secara konsisten oleh generator *deepfake* tanpa meninggalkan ketidaksesuaian visual, sehingga model cenderung lebih konsisten dalam membentuk batas keputusan (*decision boundary*) pada kelas laki-laki.

### 5.7.3. Visualisasi Hasil Prediksi Data Uji

Sebagai pelengkap dari evaluasi metrik statistik, dilakukan inspeksi visual terhadap beberapa sampel prediksi pada data uji. Visualisasi ini bertujuan untuk menunjukkan hasil klasifikasi model EfficientNet-B0 pada citra wajah secara individual beserta nilai probabilitas prediksinya. Melalui pendekatan ini, peneliti dapat melihat secara langsung bagaimana model memproses karakteristik fisik wajah dari suatu gambar dan membandingkannya dengan tebakan akhirnya.

Sebagai langkah awal, program diinstruksikan untuk mengambil 12 gambar secara acak dari kumpulan data uji. Pengambilan sampel acak tanpa pengembalian (*without replacement*) ini berfungsi untuk memastikan bahwa gambar yang ditampilkan benar-benar bervariasi dan mampu mewakili kondisi data yang sesungguhnya tanpa adanya rekayasa pemilihan.

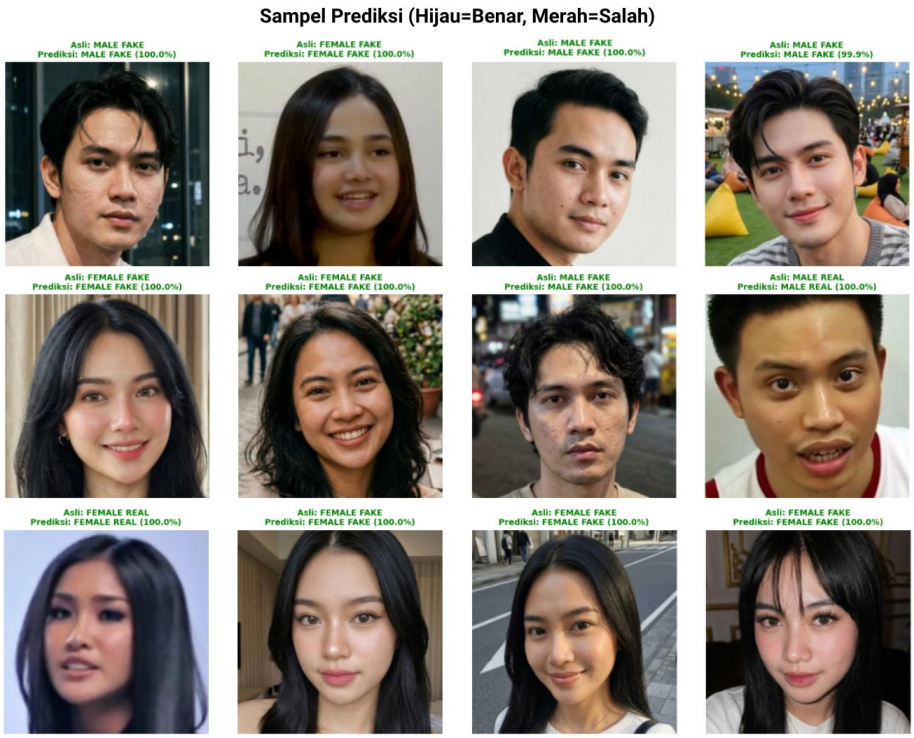



**Gambar 5.15.** Proses Menampilkan Hasil Prediksi

Sebelum gambar sampel tersebut diproses untuk mendapatkan hasil prediksi, sistem perlu melakukan penyesuaian dimensi matriks secara terprogram. Arsitektur komputasi bawaan PyTorch dirancang untuk menerima tumpukan data dalam format empat dimensi, sehingga gambar tunggal yang dipilih harus diberikan tambahan dimensi bayangan (*unsqueeze*) agar kompatibel dengan lapisan masukan model. Setelah gambar berhasil diproses melintasi seluruh lapisan jaringan, nilai keluaran yang dihasilkan kemudian dihitung menggunakan fungsi aktivasi *Softmax*. Penggunaan perhitungan *Softmax* pada tahapan analisis ini sangat krusial karena ia bertugas mengubah nilai keluaran mentah model menjadi skala probabilitas. Melalui nilai probabilitas ini, sistem dapat mengekstraksi tingkat keyakinan (*confidence score*) dari model, sehingga peneliti dapat mengetahui seberapa yakin arsitektur tersebut terhadap tebakannya sendiri dalam persentase nol hingga seratus persen.

Setelah data hasil prediksi dan tingkat keyakinan berhasil diekstraksi, langkah pengolahan terakhir adalah menampilkan gambar asli beserta keterangan prediksinya ke dalam format grafik. Karena seluruh gambar latih dan uji sebelumnya telah diubah rentang pikselnya melalui proses normalisasi standar ImageNet, gambar tersebut akan terlihat gelap dan tidak wajar jika langsung dicetak. Oleh karena itu, sistem menerapkan prosedur pembalikan normalisasi dengan mengalikan matriks gambar terhadap nilai standar deviasi dan menambahkan nilai rata-rata aslinya, sehingga warna gambar kembali seperti kondisi semula yang dapat dikenali oleh mata manusia. Kedua belas gambar yang

telah diperbaiki tersebut kemudian disusun rapi ke dalam tata letak berukuran 3x4. Untuk mempermudah proses identifikasi secara visual, setiap gambar dilengkapi dengan keterangan label asli, hasil prediksi, serta nilai persentase keyakinan yang diwarnai secara dinamis. Keterangan teks akan berwarna hijau apabila model memberikan tebakan yang benar dan otomatis berubah menjadi merah apabila terjadi kesalahan klasifikasi pada gambar tersebut. Representasi visual sampel prediksi tersebut disajikan pada Gambar 5.16.




**Gambar 5.16.** Visualisasi Sampel Hasil Prediksi Data Uji

Berdasarkan visualisasi pada Gambar 5.16, ditampilkan 12 sampel citra acak dari data pengujian. Setiap citra dilengkapi anotasi yang membandingkan label kelas asli dengan label prediksi model. Seluruh anotasi ditampilkan dengan teks berwarna hijau, yang menunjukkan bahwa model berhasil mengklasifikasikan seluruh sampel tersebut dengan benar (*true positives* dan *true negatives*). Visualisasi ini menunjukkan tingginya nilai probabilitas prediksi (*confidence score*) pada setiap sampel. Model menghasilkan probabilitas antara 99.9% hingga 100.0% pada seluruh citra yang ditampilkan, baik untuk kelas *fake* maupun *real*. Nilai probabilitas yang tinggi ini konsisten dengan hasil evaluasi *Area Under Curve* (AUC) sebelumnya yang berada pada rentang 0.999–1.000. Hasil tersebut

menunjukkan bahwa model memiliki batas keputusan (*decision boundary*) yang jelas dalam membedakan citra wajah asli dan *deepfake*.

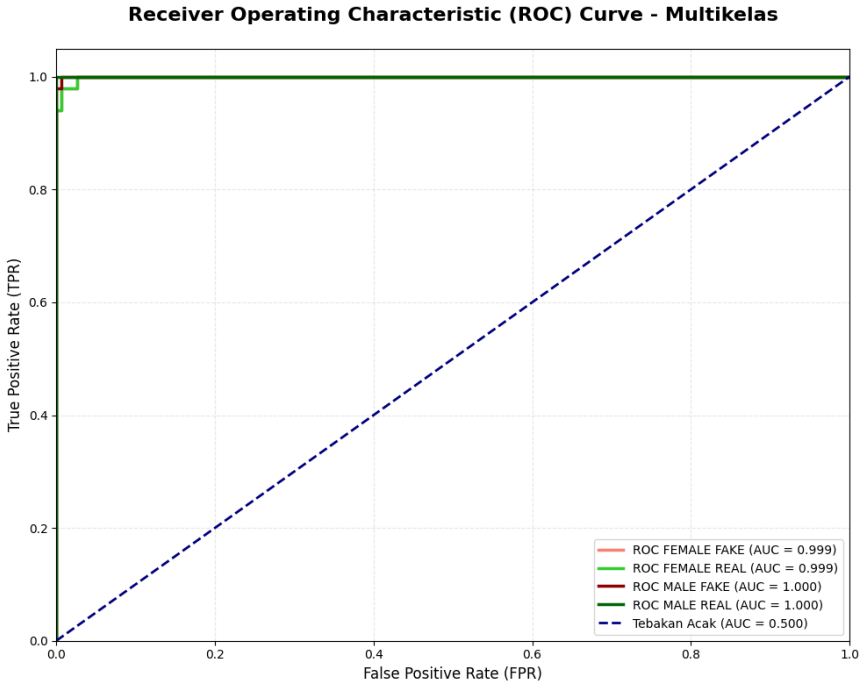
Lebih lanjut, variasi elemen visual pada sampel citra menunjukkan bahwa model mampu mempertahankan kinerja klasifikasi pada berbagai kondisi. Sampel pada Gambar 5.16 mencakup variasi pencahayaan, pose, resolusi, serta latar belakang yang beragam seperti latar polos, ruangan *indoor*, dan lingkungan terbuka. Konsistensi hasil prediksi pada kondisi tersebut mengindikasikan bahwa model mampu memfokuskan proses ekstraksi fitur pada area wajah yang relevan, dengan pengaruh yang relatif rendah dari elemen non-wajah di sekitarnya.

### 5.7.4. Kurva ROC dan Skor AUC Multikelas

Pengujian kinerja model dilanjutkan melalui analisis probabilitas prediksi menggunakan kurva *Receiver Operating Characteristic* (ROC) serta metrik *Area Under the Curve* (AUC). Kurva ROC merupakan metode evaluasi grafis yang menggambarkan hubungan antara *True Positive Rate* (TPR) atau sensitivitas pada sumbu vertikal (Y) dan *False Positive Rate* (FPR) pada sumbu horizontal (X) pada berbagai nilai ambang (t*hreshold*). Karena model dalam penelitian ini menangani klasifikasi multikelas (empat kategori wajah), digunakan pendekatan *One-vs-Rest* (OvR). Melalui pendekatan ini, kinerja model dievaluasi secara terpisah untuk setiap kelas dengan membandingkan satu kelas sebagai positif terhadap gabungan kelas lainnya sebagai negatif. Luas area di bawah kurva kemudian dinyatakan dalam bentuk skor AUC dengan rentang nilai 0 hingga 1.

Setelah perhitungan matematis selesai, hasil evaluasi divisualisasikan ke dalam bentuk grafik dua dimensi. Pada grafik tersebut, sumbu mendatar mewakili tingkat kesalahan prediksi, sedangkan sumbu tegak mewakili tingkat keberhasilan prediksi. Grafik visualisasi ini juga dilengkapi dengan garis putus-putus diagonal yang melambangkan batas kinerja tebakan acak dengan nilai AUC sebesar 0.500. Kinerja model yang optimal ditandai dengan kurva menjauhi garis putus-putus tersebut menuju sudut kiri atas grafik, yang berarti nilai AUC semakin mendekati angka sempurna yaitu 1.000. Semakin tinggi nilai AUC yang diraih oleh masing-masing kelas, semakin tangguh pula kemampuan arsitektur model dalam memisahkan gambar wajah asli dan *deepfake* tanpa memunculkan banyak

kesalahan identifikasi. Visualisasi kurva ROC dan skor AUC disajikan pada Gambar 5.17.




**Gambar 5.17.** Kurva ROC dan Skor AUC Multikelas

Berdasarkan visualisasi pada Gambar 5.17, grafik menampilkan garis diagonal putus-putus yang membentang dari sudut kiri bawah ke kanan atas sebagai garis acuan (*baseline*) yang merepresentasikan klasifikasi acak dengan nilai AUC sebesar 0.50. Dalam analisis ROC, kinerja model dinilai semakin baik apabila kurva mendekati sudut kiri atas grafik (FPR mendekati 0 dan TPR mendekati 1), yang menunjukkan kemampuan pemisahan kelas yang tinggi. Hasil pemetaan menunjukkan bahwa kurva dari keempat kelas saling berhimpitan dan cenderung mengikuti batas atas serta sisi kiri grafik. Pola ini mengindikasikan bahwa model EfficientNet-B0 memiliki kemampuan yang baik dalam membedakan kelas serta mampu menekan nilai *false positive* pada tingkat yang rendah.

Rincian skor AUC pada legenda grafik menunjukkan kemampuan model dalam membedakan fitur antar kelas. Kelas *male fake* dan *male real* memperoleh nilai AUC sebesar 1.000, yang mengindikasikan kemampuan pemisahan kelas yang sangat baik pada kategori tersebut. Sementara itu, kelas *female fake* dan *female real* memiliki nilai AUC sebesar 0.999, yang menunjukkan kinerja yang juga sangat tinggi meskipun tidak mencapai nilai maksimum. Selisih kecil sebesar 0.001 pada kategori perempuan ini sejalan dengan temuan pada analisis *confusion matrix* dan

*classification report* sebelumnya. Perbedaan tersebut mengindikasikan adanya sejumlah kecil citra wajah perempuan yang berada di sekitar batas keputusan (*decision boundary*) dalam ruang fitur model, sehingga berkontribusi terhadap terjadinya beberapa kesalahan prediksi.

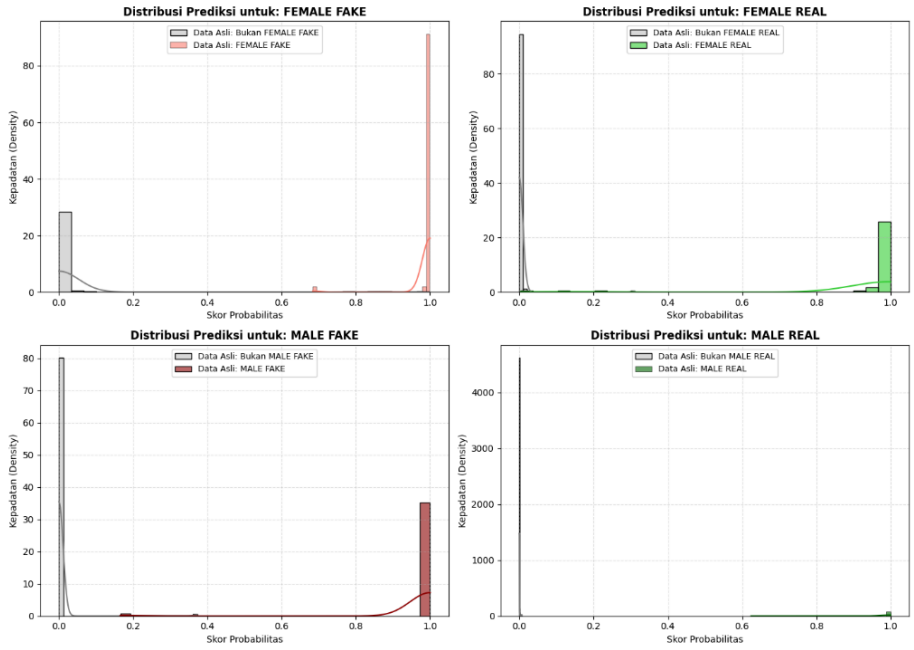
Meskipun terdapat sedikit penurunan pada kelas perempuan, seluruh nilai AUC yang berada di atas 0.99 menunjukkan bahwa model memiliki kemampuan diskriminasi yang sangat baik dalam membedakan antar kelas. Hasil evaluasi ini mendukung temuan sebelumnya bahwa tingginya nilai akurasi yang dicapai model berkaitan dengan kemampuan arsitektur dalam merepresentasikan pola spasial wajah dan artefak *deepfake*, bukan akibat prediksi yang bersifat acak.

### 5.8. Analisis Kepercayaan Keputusan (*Confidence*)

Evaluasi keandalan model klasifikasi tidak hanya didasarkan pada akurasi, tetapi juga mempertimbangkan tingkat keyakinan (*confidence*) dalam setiap prediksi. Analisis distribusi probabilitas prediksi bertujuan untuk mengukur seberapa besar tingkat keyakinan model saat mengklasifikasikan sebuah gambar, bukan sekadar melihat status benar atau salah dari hasil tebakan akhirnya. Analisis ini dilakukan dengan meninjau distribusi probabilitas keluaran yang dihasilkan oleh fungsi aktivasi *Softmax* pada lapisan akhir model. Melalui pendekatan ini, dapat diamati apakah prediksi didasarkan pada pemisahan fitur yang jelas atau berada pada kondisi ketidakpastian (probabilitas mendekati 0.5).

Untuk memberikan wawasan analisis yang komprehensif pada skenario klasifikasi empat kelas, sistem membedah dan memisahkan skor probabilitas tersebut secara spesifik untuk masing-masing kelas target. Pada setiap kelas yang dievaluasi, data probabilitas dibagi menjadi dua kelompok observasi yang berbeda. Kelompok pertama merupakan kumpulan probabilitas dari gambar yang label aslinya memang sesuai dengan kelas target yang sedang diukur. Sebaliknya, kelompok kedua merupakan probabilitas dari gambar yang label aslinya berasal dari ketiga kelas lainnya. Proses pemisahan komputasional ini memiliki peran yang sangat penting untuk memastikan apakah model mampu memberikan skor keyakinan yang tinggi secara eksklusif hanya pada kelas yang tepat dan memberikan skor yang rendah pada kelas yang salah.

Hasil pemisahan kelompok observasi tersebut kemudian divisualisasikan ke dalam empat grafik matriks histogram yang dilengkapi dengan kurva estimasi kepadatan (*Kernel Density Estimation* atau KDE). Pada masing-masing grafik, distribusi probabilitas dari kelompok data yang bukan merupakan kelas target direpresentasikan dengan area berwarna abu-abu, sedangkan kelompok data yang sesuai dengan kelas target direpresentasikan dengan warna yang spesifik. Indikator kinerja model yang sangat optimal ditandai dengan adanya pemisahan yang tegas antara kedua distribusi tersebut. Kurva berwarna spesifik idealnya menumpuk dan memuncak di sisi kanan grafik yang mendekati angka probabilitas 1.0, sementara kurva berwarna abu-abu menumpuk di sisi kiri grafik yang mendekati angka 0.0. Apabila persimpangan atau area tumpang antara kedua kurva relatif kecil, maka model memiliki kemampuan yang baik dalam membedakan fitur antar kelas serta menunjukkan tingkat ambiguitas klasifikasi yang rendah. Visualisasi distribusi probabilitas untuk keempat kelas target disajikan pada Gambar 5.18.




**Gambar 5.18.** Visualisasi Distribusi Probabilitas Prediksi

Berdasarkan Gambar 5.18, visualisasi disajikan dalam empat grafik yang masing-masing merepresentasikan satu kelas. Sumbu horizontal menunjukkan skor probabilitas pada rentang 0.0 (bukan kelas target) hingga 1.0 (kelas target), sedangkan sumbu vertikal menunjukkan kepadatan distribusi (*density*). Pada model *deep learning* yang terlatih dengan baik, distribusi yang diharapkan cenderung

bersifat bimodal, di mana data terkonsentrasi pada dua kutub ekstrem dan relatif sedikit berada di area tengah.

Hasil pemetaan menunjukkan bahwa arsitektur EfficientNet-B0 membentuk distribusi yang mendekati pola bimodal. Analisis terhadap data yang bukan termasuk dalam kelas target (direpresentasikan oleh histogram berwarna abu-abu) menunjukkan konsentrasi kepadatan yang tinggi di sekitar probabilitas 0.0. Pola ini mengindikasikan bahwa model memiliki kemampuan penolakan (*rejection capability*) yang baik, di mana citra yang tidak memiliki karakteristik suatu kelas cenderung diberikan nilai probabilitas yang sangat rendah.

Sementara itu, distribusi pada data yang termasuk dalam masing-masing kelas (direpresentasikan oleh histogram berwarna merah, hijau terang, merah tua, dan hijau tua) menunjukkan konsentrasi kepadatan yang tinggi di sekitar probabilitas 1.0. Hal ini mengindikasikan bahwa ketika model menghasilkan prediksi yang benar, keputusan tersebut umumnya disertai dengan tingkat keyakinan yang tinggi. Kondisi ini menunjukkan bahwa model mampu mengidentifikasi pola visual yang sesuai dengan representasi fitur yang dipelajari selama proses pelatihan.

Jika dianalisis lebih lanjut, terdapat pola pada grafik *female real* (kanan atas) yang konsisten dengan hasil evaluasi sebelumnya. Berbeda dengan kelas laki-laki yang distribusinya terkonsentrasi di sekitar probabilitas 1.0, histogram pada kelas *female real* menunjukkan penyebaran yang sedikit melebar hingga mendekati probabilitas 0.8, serta adanya ekor distribusi yang menjauh dari nilai maksimum. Pola ini sejalan dengan penurunan nilai pada *classification report* dan munculnya tiga kesalahan klasifikasi pada *confusion matrix*. Distribusi tersebut mengindikasikan bahwa pada beberapa citra wajah perempuan asli, model memberikan tingkat keyakinan yang sedikit lebih rendah. Hal ini kemungkinan dipengaruhi oleh variasi visual seperti riasan atau kondisi pencahayaan yang memengaruhi representasi fitur yang dipelajari model, meskipun sebagian besar prediksi tetap sesuai dengan label sebenarnya.

Secara keseluruhan, minimnya distribusi probabilitas pada rentang 0.2 hingga 0.8 di seluruh kelas menunjukkan bahwa model jarang menghasilkan prediksi dengan tingkat keyakinan menengah. Pola ini mengindikasikan bahwa model memiliki tingkat ambiguitas klasifikasi yang rendah. Dengan demikian, arsitektur

EfficientNet-B0 mampu membentuk batas keputusan (*decision boundary*) yang jelas, sehingga prediksi terhadap citra wajah asli maupun manipulasi *deepfake* umumnya disertai dengan tingkat keyakinan yang tinggi.

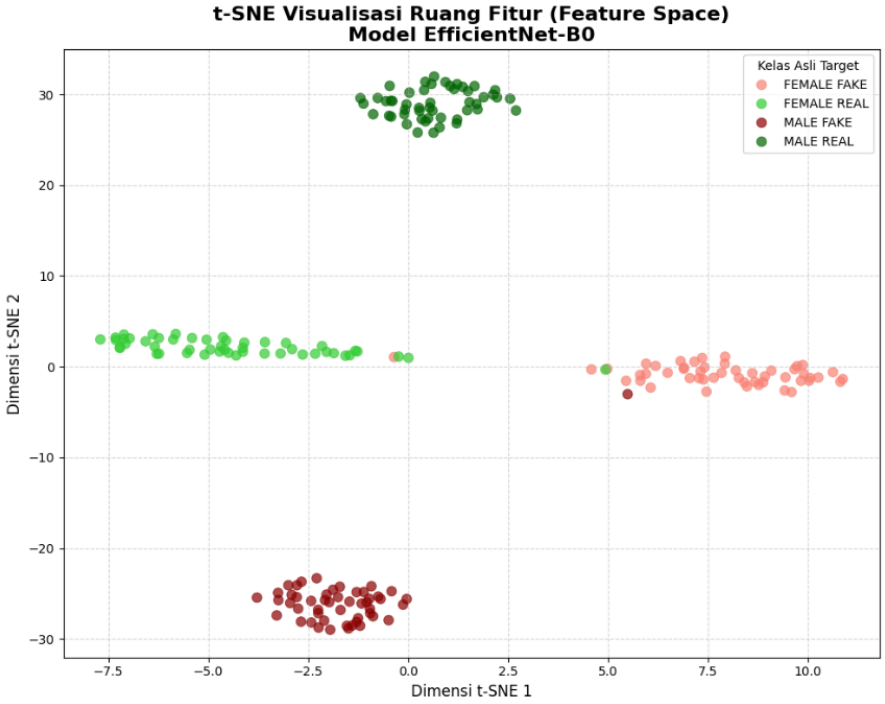
## 5.9. Interpretasi Model (*Explainable AI*)

Pendekatan *Explainable Artificial Intelligence* (XAI) digunakan untuk meningkatkan transparansi dalam memahami cara kerja arsitektur EfficientNet-B0 yang bersifat *black-box*. Salah satu metode yang digunakan adalah visualisasi ruang fitur (*feature space*) melalui algoritma *t-Distributed Stochastic Neighbor Embedding* (t-SNE). Visualisasi ruang fitur menggunakan metode algoritma t-SNE bertujuan untuk membongkar kotak hitam algoritma dan mengamati bagaimana model menyusun representasi data secara internal sebelum membuat keputusan klasifikasi akhir. Proses pengujian diawali dengan ekstraksi fitur dari seluruh citra pada data uji. Pada tahap ini, sistem mengambil keluaran dari lapisan konvolusi utama arsitektur EfficientNet-B0 sebelum proses klasifikasi akhir dilakukan. Setelah melewati lapisan *global average pooling*, representasi fitur diratakan (*flattening*) sehingga menghasilkan vektor fitur berdimensi 1280 untuk setiap citra wajah.

Mengingat kumpulan fitur tersebut berada dalam ruang berdimensi 1280 yang mustahil untuk divisualisasikan secara langsung, sistem memerlukan teknik komputasi untuk mereduksi dimensi tersebut. Oleh karena itu, penelitian ini mengimplementasikan algoritma reduksi dimensi matematis t-SNE. Algoritma ini dirancang secara khusus untuk memadatkan data dari ruang berdimensi sangat tinggi menjadi koordinat dua dimensi sederhana, dengan cara mempertahankan struktur kedekatan spasial dari data aslinya. Mekanisme transformasi matematis ini memastikan bahwa gambar-gambar yang dianggap memiliki kemiripan fitur visual oleh model akan diletakkan saling berdekatan, sedangkan gambar yang memiliki fitur berbeda akan diposisikan saling berjauhan pada ruang koordinat yang baru.

Setelah proses reduksi dimensi diselesaikan oleh komputer, koordinat dua dimensi yang dihasilkan langsung divisualisasikan ke dalam bentuk grafik titik sebar (*scatter plot*). Untuk memudahkan pengamatan pola pengelompokan, setiap titik data diberikan identitas warna yang spesifik berdasarkan label kebenaran dasarnya. Kategori wajah rekayasa divisualisasikan menggunakan gradasi warna

merah, sedangkan kategori wajah asli direpresentasikan menggunakan gradasi warna hijau. Melalui peta sebaran titik ini, kemampuan model dalam membedakan fitur citra dapat dilihat secara lebih mudah. Arsitektur jaringan yang berfungsi dengan optimal akan menghasilkan kumpulan titik yang memusat berdasarkan warnanya masing-masing dan memiliki garis batas pemisah yang jelas antar kelas. Hal ini membuktikan bahwa model tidak menebak secara acak, melainkan benar-benar memahami perbedaan struktur intrinsik antara wajah manusia asli dan *deepfake*. Hasil visualisasi ruang fitur disajikan pada Gambar 5.19.




**Gambar 5.19.** Visualisasi Ruang Fitur

Berdasarkan visualisasi pada Gambar 5.19, terlihat bahwa model memetakan representasi fitur ke dalam empat klaster utama yang relatif terpisah. Setiap titik pada grafik merepresentasikan satu citra dari data uji. Jarak spasial antar kelompok, terutama pada sumbu vertikal (dimensi t-SNE 2) antara kelas *male real* (titik hijau tua di bagian atas) dan *male fake* (titik merah tua di bagian bawah), menunjukkan adanya pemisahan fitur yang jelas pada kategori wajah laki-laki. Selain itu, kerapatan titik pada kedua klaster laki-laki sejalan dengan hasil evaluasi sebelumnya yang menunjukkan tingkat akurasi tinggi serta minimnya kesalahan klasifikasi pada *confusion matrix* dan kurva ROC.

Sementara itu, pada sumbu horizontal (dimensi t-SNE 1), kelas *female real* (titik hijau muda) dan *female fake* (titik merah muda) terletak pada area tengah dengan posisi yang saling berseberangan. Klaster wajah perempuan menunjukkan sebaran yang relatif lebih luas dibandingkan dengan klaster wajah laki-laki. Pola ini mengindikasikan adanya variasi fitur intrakelompok yang lebih tinggi pada data wajah perempuan, sehingga model perlu merepresentasikan rentang karakteristik visual yang lebih beragam.

Salah satu temuan penting dari visualisasi t-SNE ini adalah kemampuannya dalam menjelaskan anomali prediksi secara lebih transparan. Terlihat beberapa titik data yang berada di luar klaster utamanya (*outlier*), di antaranya satu titik *male fake* (merah tua) dan beberapa titik *female real* (hijau muda) yang bergeser mendekati klaster *female fake* (merah muda). Pola ini sejalan dengan hasil *confusion matrix*, di mana model salah mengklasifikasikan satu citra *male fake* dan tiga citra *female real* sebagai *female fake*. Dalam ruang fitur berdimensi tinggi, kedekatan posisi titik-titik tersebut mengindikasikan adanya kemiripan karakteristik visual yang direpresentasikan secara matematis oleh model. Akibatnya, pada tahap klasifikasi akhir beberapa data tersebut berada di sekitar batas keputusan (*decision boundary*), sehingga berpotensi mengalami kesalahan klasifikasi.

Visualisasi t-SNE ini memberikan dukungan empiris terhadap kinerja model. Jarak antar klaster yang relatif terpisah serta pola pengelompokan yang jelas menunjukkan bahwa model tidak melakukan klasifikasi secara acak maupun sekadar menghafal data latih (*memorization*). Sebaliknya, model mampu merepresentasikan atribut visual yang relevan, seperti tekstur kulit, pola pencahayaan, dan artefak manipulasi, yang berperan dalam membedakan antara wajah asli dan *deepfake*.

### 5.10. Analisis Kesalahan Model

Evaluasi kinerja model klasifikasi kecerdasan buatan perlu dilengkapi dengan analisis terhadap data yang mengalami kesalahan prediksi. *Error analysis* bertujuan untuk mengidentifikasi keterbatasan arsitektur model, meninjau potensi bias, serta memahami penyebab kesalahan secara visual. Berdasarkan hasil pengujian pada 200 data uji, model EfficientNet-B0 memperoleh tingkat akurasi sebesar 98% dengan empat citra yang mengalami kesalahan klasifikasi (*misclassification*).

Kumpulan citra yang tidak terklasifikasi dengan tepat tersebut disajikan pada Gambar 5.20 untuk dianalisis lebih lanjut.


**Gambar 5.20.** Katalog Kesalahan Klasifikasi

Berdasarkan inspeksi visual pada Gambar 5.20, kesalahan prediksi model dapat diklasifikasikan ke dalam tiga faktor utama, yaitu ilusi kosmetik, kemiripan fitur gender, dan degradasi kualitas citra. Pertama, kesalahan yang berkaitan dengan ilusi kosmetik dan kehalusan tekstur terlihat pada citra pertama dan kedua, di mana label *female real* diprediksi sebagai *female fake* dengan tingkat keyakinan yang tinggi (99.7% dan 87.6%). Pada citra pertama, penggunaan riasan (*makeup*) yang tebal serta kontras pencahayaan yang kuat menyebabkan tekstur alami kulit seperti pori-pori menjadi kurang terlihat. Kondisi serupa juga terdapat pada citra kedua, yang menunjukkan pencahayaan datar (*flat lighting*) dengan tekstur kulit yang tampak seragam. Dalam konteks ini, model cenderung mengaitkan tekstur kulit yang sangat halus (*oversmoothed*) dengan artefak penghalusan yang umum dihasilkan oleh algoritma *deepfake*. Hal ini mengindikasikan bahwa model memiliki sensitivitas yang tinggi terhadap karakteristik kehalusan tekstur pada citra wajah perempuan.

Kedua, kesalahan yang berkaitan dengan kemiripan fitur gender (*gender misclassification*) terlihat pada citra ketiga, di mana citra dengan label *male fake* diklasifikasikan sebagai *female fake* dengan probabilitas 83.4%. Pada kasus ini, model telah mengidentifikasi citra sebagai hasil manipulasi (*fake*), namun kurang tepat dalam menentukan kategori gender. Hal ini diduga dipengaruhi oleh karakteristik visual subjek, seperti garis rahang yang relatif halus, gaya rambut yang membingkai wajah, serta tekstur kulit yang tampak sangat mulus akibat proses *deepfake*. Kombinasi fitur yang cenderung androgini tersebut menyebabkan

representasi fitur citra berada lebih dekat dengan klaster wajah perempuan dalam ruang fitur model, sehingga memengaruhi hasil klasifikasi pada kategori gender.

Ketiga, kesalahan yang berkaitan dengan degradasi kualitas citra terlihat pada citra keempat (paling kanan), di mana citra *female real* diklasifikasikan sebagai *female fake* dengan probabilitas 77.9%. Berbeda dengan kasus riasan, kondisi ini dipengaruhi oleh kualitas gambar yang rendah serta fokus yang kurang tajam (*soft focus* atau *blurry*). Citra yang buram menyebabkan berkurangnya detail frekuensi tinggi, seperti ketajaman rambut, tekstur kulit, dan pantulan cahaya pada area wajah. Dalam perspektif model konvolusi, penurunan kualitas visual tersebut memiliki kemiripan dengan distorsi atau artefak kompresi yang sering muncul pada *deepfake* berkualitas rendah. Kemiripan ini dapat memengaruhi representasi fitur yang dihasilkan model, sehingga sebagian citra dengan kualitas rendah cenderung diklasifikasikan sebagai hasil manipulasi.

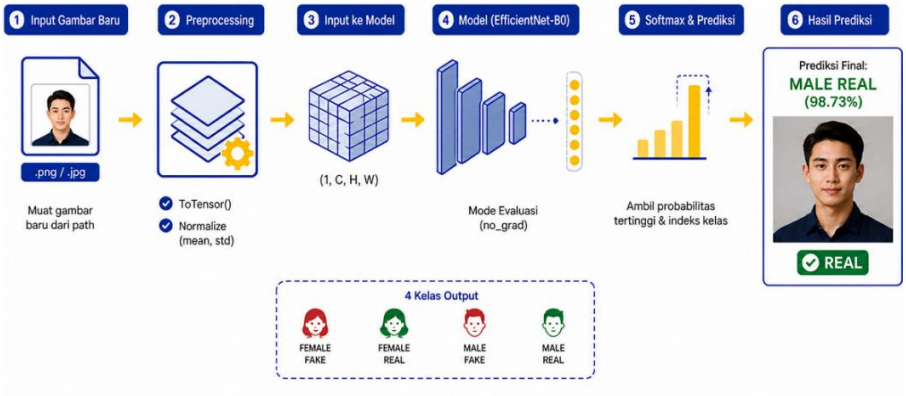
Analisis kesalahan ini memberikan wawasan bahwa meskipun model menunjukkan kinerja yang tinggi, terdapat tantangan dalam membedakan antara modifikasi wajah fisik (seperti riasan atau pencahayaan) dan manipulasi digital (*deepfake*), khususnya pada kondisi citra yang tidak ideal. Temuan ini menunjukkan bahwa variasi visual pada data nyata dapat memengaruhi representasi fitur model dan berpotensi menimbulkan kesalahan klasifikasi pada kasus tertentu.

### 5.11. Pengujian Data Baru dan Grad-CAM

Tahap akhir evaluasi kinerja model dilakukan dengan menguji ketangguhan (*robustness*) dan kemampuan generalisasi arsitektur EfficientNet-B0 pada citra baru yang berada di luar distribusi data pelatihan, validasi, dan pengujian. Pengujian menggunakan data *in-the-wild* ini bertujuan untuk menilai kinerja model pada kondisi yang lebih representatif terhadap situasi nyata. Selain itu, untuk meninjau dasar pengambilan keputusan model, digunakan visualisasi *Gradient-weighted Class Activation Mapping* (Grad-CAM).

Proses pengujian data baru diawali dengan tahapan persiapan dan prapemrosesan gambar tunggal. Secara teknis, sistem terlebih dahulu diprogram untuk memvalidasi keberadaan berkas gambar pada direktori penyimpanan yang telah ditentukan. Setelah berkas berhasil ditemukan, gambar tersebut dibuka dan secara otomatis dikonversi ke dalam ruang warna RGB (*Red, Green, Blue*). Proses

konversi ini sangat penting untuk memastikan konsistensi format saluran warna, khususnya untuk mengantisipasi gambar berformat PNG yang sering kali memiliki saluran transparansi tambahan (*Alpha channel*) yang tidak dapat diproses oleh model. Selanjutnya, gambar mentah tersebut diubah menjadi matriks tensor matematis dan dinormalisasi menggunakan parameter standar yang sama persis dengan fase validasi. Mengingat arsitektur PyTorch dirancang untuk memproses data dalam bentuk kelompok (*batch*), matriks gambar tunggal ini juga diberikan penambahan dimensi bayangan agar struktur datanya selaras dengan format masukan yang diharapkan oleh lapisan jaringan.

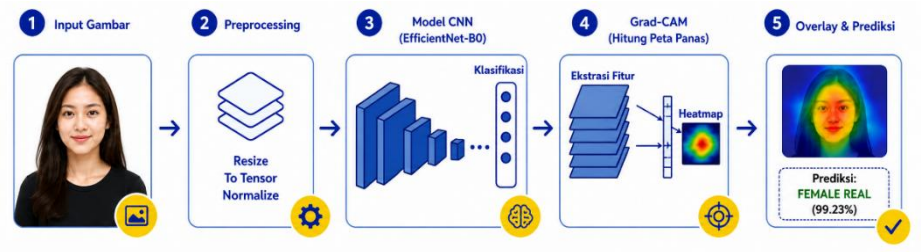



**Gambar 5.21.** Proses Prediksi Data Baru

Setelah persiapan data selesai, sistem beralih pada tahap eksekusi prediksi. Model diatur ke dalam mode evaluasi dan fungsi pelacakan gradien matematis dinonaktifkan sepenuhnya demi efisiensi memori komputer. Matriks gambar kemudian disalurkan ke dalam arsitektur EfficientNet-B0 untuk diekstraksi fitur visualnya. Nilai prediksi mentah (*logits*) yang dihasilkan oleh model kemudian diproses menggunakan fungsi aktivasi *Softmax*. Fungsi ini bertugas mengonversi nilai mentah tersebut menjadi distribusi probabilitas dengan rentang nol hingga seratus persen untuk keempat kelas target. Dari perhitungan probabilitas tersebut, sistem akan menyaring dan mengambil nilai persentase tertinggi beserta label kelas yang bersesuaian sebagai hasil keputusan final dari model. Selanjutnya, hasil prediksi divisualisasikan ke dalam bentuk antarmuka grafik agar mudah dipahami oleh pengguna. Gambar wajah yang diuji ditampilkan secara utuh tanpa pemotongan, didampingi dengan keterangan teks yang memuat label prediksi akhir dan skor persentase keyakinan model. Untuk mempercepat proses identifikasi secara visual, sistem juga menerapkan logika pewarnaan dinamis pada teks

keterangan tersebut. Apabila model menyimpulkan bahwa gambar tersebut adalah wajah asli (mengandung kata *real*), maka teks akan berwarna hijau. Sebaliknya, apabila gambar tersebut terdeteksi sebagai wajah hasil rekayasa (mengandung kata *fake*), teks peringatan akan otomatis berubah menjadi warna merah.

Tahap berikutnya dalam skenario pengujian data baru adalah penerapan metode Grad-CAM atau *Gradient-weighted Class Activation Mapping*. Evaluasi ini bertujuan untuk memberikan transparansi terhadap proses pengambilan keputusan oleh model klasifikasi. Meskipun model telah berhasil memberikan hasil prediksi berupa probabilitas persentase yang tinggi, peneliti tetap perlu mengetahui area spesifik mana dari gambar wajah yang paling memengaruhi keputusan komputasi tersebut. Secara teknis, implementasi Grad-CAM pada penelitian ini difokuskan secara khusus pada lapisan ekstraksi fitur konvolusi terakhir dari arsitektur EfficientNet-B0. Pemilihan lapisan akhir ini didasarkan pada karakteristik hierarki jaringan saraf tiruan, di mana lapisan terdalam mampu menangkap representasi visual yang paling kompleks dan kaya akan informasi semantik spasial sebelum data disalurkan ke lapisan klasifikasi akhir.




**Gambar 5.22.** Alur Kerja Grad-CAM

Proses visualisasi diawali dengan memuat gambar uji dan melakukan prapemrosesan piksel berupa konversi tensor dan normalisasi agar sejalan dengan format yang dapat dibaca oleh algoritma. Setelah gambar dipersiapkan, sistem akan menyalurkannya ke dalam model untuk mendapatkan nilai keluaran prediksi. Algoritma Grad-CAM kemudian bekerja dengan cara melacak balik perhitungan gradien matematis dari kelas target yang diprediksi menuju lapisan konvolusi terakhir yang telah ditetapkan sebelumnya. Perhitungan turunan ini menghasilkan sebuah peta aktivasi dua dimensi yang mengukur seberapa besar bobot kontribusi setiap wilayah spasial terhadap hasil klasifikasi akhir. Peta aktivasi tersebut pada awalnya berbentuk matriks skala abu-abu (*grayscale*), yang kemudian secara

terprogram dikonversi menjadi format peta warna termal (*heatmap*) untuk mempermudah proses identifikasi oleh mata manusia.

Langkah terakhir dari tahapan ini adalah memproyeksikan peta warna termal tersebut tepat di atas gambar wajah aslinya. Pada visualisasi yang dihasilkan, area gambar yang direpresentasikan dengan warna hangat seperti merah atau kuning menunjukkan titik fokus utama yang paling memicu model untuk mengenali gambar tersebut sebagai wajah asli atau *deepfake*. Sebaliknya, area yang ditutupi oleh warna dingin seperti biru atau ungu menunjukkan bagian piksel yang diabaikan atau dianggap tidak memiliki fitur yang relevan oleh model. Hasil proyeksi ini kemudian ditampilkan secara berdampingan dengan gambar aslinya di dalam satu bingkai grafik. Melalui evaluasi visual ini keandalan model tidak hanya diukur melalui persentase metrik akurasi semata, melainkan juga dapat diverifikasi secara logis melalui kesesuaian pola fitur wajah yang dianalisis oleh sistem. Hasil pengujian pada empat sampel citra beserta peta aktivasi visualnya disajikan pada Gambar 5.23.


**Gambar 5.23.** Hasil Prediksi Data Baru dan Grad-CAM

### 5.11.1. *Model Robustness*

Berdasarkan visualisasi pada Gambar 5.23, model diuji menggunakan empat sampel citra baru yang mewakili seluruh kelas target (*female real*, *male real*, *female fake*, dan *male fake*). Citra tersebut memiliki variasi pose, pencahayaan, dan latar belakang yang tidak terdapat pada data pelatihan. Hasil pengujian menunjukkan

bahwa arsitektur EfficientNet-B0 mampu mengklasifikasikan seluruh sampel dengan tepat serta menghasilkan nilai probabilitas prediksi yang tinggi.

Pada kelas *female real* dan *male fake*, model menghasilkan probabilitas prediksi sebesar 1.00 (100%). Sementara itu, pada kelas *male real* dan *female fake*, model memprediksi label yang sesuai dengan probabilitas masing-masing sebesar 0.9978 (99.78%) dan 0.9896 (98.96%). Konsistensi hasil klasifikasi serta nilai probabilitas yang tinggi pada data baru ini menunjukkan bahwa model memiliki tingkat *robustness* yang baik. Temuan ini mengindikasikan bahwa model tidak hanya bergantung pada karakteristik spesifik dari data pelatihan, tetapi juga mampu merepresentasikan pola umum yang membedakan antara wajah asli dan manipulasi *deepfake*.

### 5.11.2. Interpretasi Grad-CAM

Pendekatan Grad-CAM digunakan untuk memvisualisasikan kontribusi area citra terhadap hasil klasifikasi dengan menghasilkan peta panas (*heatmap*) pada lapisan konvolusi terakhir. Pada Gambar 5.23, area dengan spektrum warna hangat (merah, jingga, dan kuning) menunjukkan bagian citra yang memiliki pengaruh terbesar terhadap keputusan model, sedangkan warna dingin (biru gelap) menunjukkan area dengan kontribusi yang lebih rendah. Berdasarkan analisis visual peta Grad-CAM pada keempat sampel, terlihat bahwa model memberikan perhatian utama pada area tekstur kulit wajah, terutama di bagian pipi, sekitar hidung (*T-zone*), dan garis rahang. Pola atensi ini sejalan dengan karakteristik umum dalam deteksi *deepfake*, di mana area tersebut sering mengalami modifikasi seperti ketidaksesuaian warna (*color blending*) dan artefak penghalusan (*smoothing artifacts*).

Lebih lanjut, *heatmap* Grad-CAM menunjukkan bahwa model memusatkan perhatian pada area yang relevan dalam proses klasifikasi. Area dengan warna biru gelap umumnya muncul pada bagian tepi gambar, yang mengindikasikan bahwa kontribusi latar belakang terhadap keputusan model relatif rendah. Model cenderung tidak dipengaruhi oleh elemen non-wajah seperti latar belakang atau atribut pakaian. Pola atensi yang terfokus pada tekstur wajah serta minimnya pengaruh dari elemen distraksi menunjukkan bahwa model memanfaatkan fitur yang relevan dalam membedakan citra asli dan manipulasi *deepfake*. Hal ini

mengindikasikan bahwa arsitektur klasifikasi memiliki tingkat keandalan yang baik dalam mendeteksi manipulasi citra wajah.

## 5.12. Implementasi Sistem Deteksi *Deepfake* Berbasis Web

Setelah model EfficientNet-B0 selesai dilatih dan dievaluasi, tahap selanjutnya adalah mengimplementasikan model tersebut ke dalam sebuah aplikasi berbasis web. Implementasi ini bertujuan untuk menunjukkan bahwa model yang telah dikembangkan tidak hanya mampu memberikan hasil yang baik pada proses pengujian, tetapi juga dapat digunakan secara langsung oleh pengguna melalui antarmuka yang mudah dioperasikan. Aplikasi dikembangkan menggunakan *framework* Streamlit karena menyediakan mekanisme pembangunan antarmuka berbasis Python yang sederhana serta mendukung integrasi langsung dengan model *deep learning* yang dibangun menggunakan PyTorch.

Proses implementasi dilakukan dengan memanfaatkan repositori GitHub sebagai media penyimpanan kode program, kemudian aplikasi dipublikasikan menggunakan layanan *Streamlit Community Cloud*. Seluruh berkas yang diperlukan, meliputi kode program utama, definisi arsitektur EfficientNet-B0, bobot model hasil pelatihan (.pth), modul inferensi, komponen antarmuka, serta berkas pendukung lainnya disusun dalam struktur direktori yang terorganisasi sehingga memudahkan proses pemeliharaan maupun pengembangan sistem.

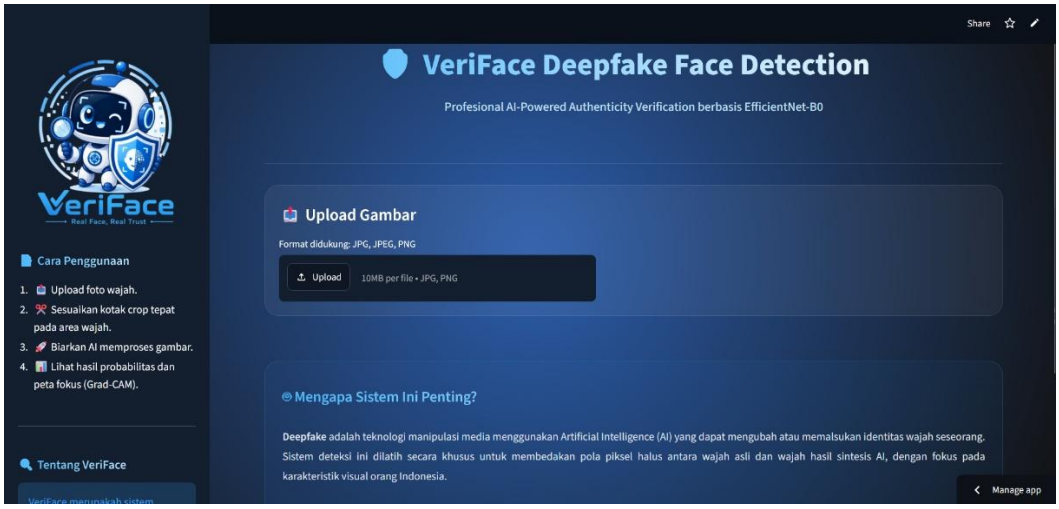
Aplikasi dibangun secara modular dengan memisahkan setiap fungsi ke dalam beberapa berkas sesuai tanggung jawabnya. Berkas *app.py* berperan sebagai pusat kendali aplikasi yang mengatur alur antarmuka, proses unggah citra, pemanggilan model, hingga penyajian hasil prediksi. Definisi arsitektur EfficientNet-B0 ditempatkan pada berkas *model_architecture.py*, sedangkan proses inferensi model, pra-pemrosesan citra, serta pembangkitan visualisasi Grad-CAM diimplementasikan pada berkas *ai_logic.py*. Komponen antarmuka, seperti *header*, kartu hasil prediksi, *sidebar*, dan informasi penggunaan, dipisahkan ke dalam *ui_components.py* sehingga kode program menjadi lebih terstruktur dan mudah dipelihara.

Untuk menjalankan aplikasi pada lingkungan *deployment*, seluruh pustaka yang dibutuhkan didefinisikan pada berkas *requirements.txt*. Berkas ini memuat berbagai dependensi, antara lain Streamlit sebagai *framework* aplikasi web,

PyTorch dan Torchvision sebagai *framework* deep learning, OpenCV dan Pillow untuk pengolahan citra, Matplotlib untuk visualisasi, serta pustaka Grad-CAM yang digunakan dalam implementasi *Explainable Artificial Intelligence* (XAI). Selain itu, tampilan antarmuka dikustomisasi menggunakan berkas CSS sehingga menghasilkan desain yang lebih modern dan meningkatkan kenyamanan pengguna saat berinteraksi dengan sistem.

### 5.12.1. Tampilan Awal *Website*

Tampilan awal aplikasi deteksi *deepfake* berbasis web ditunjukkan pada Gambar 5.24. Halaman utama dirancang dengan antarmuka yang sederhana dan mudah dipahami sehingga pengguna dapat langsung menggunakan sistem tanpa memerlukan konfigurasi tambahan. Pada bagian atas halaman ditampilkan identitas aplikasi berupa nama *VeriFace Deepfake Face Detection* beserta deskripsi singkat bahwa sistem memanfaatkan model EfficientNet-B0 sebagai mesin utama klasifikasi citra wajah.




**Gambar 5.24.** Tampilan Awal *Website*

Pada sisi kiri halaman (*sidebar*) disajikan logo aplikasi, petunjuk penggunaan, serta informasi singkat mengenai sistem. Petunjuk penggunaan disusun dalam bentuk langkah-langkah sederhana, yaitu mengunggah citra wajah, menyesuaikan area pemotongan (*crop*), menunggu proses analisis, dan melihat hasil prediksi beserta visualisasi Grad-CAM. Penyajian informasi ini bertujuan untuk memudahkan pengguna dalam memahami alur penggunaan aplikasi.

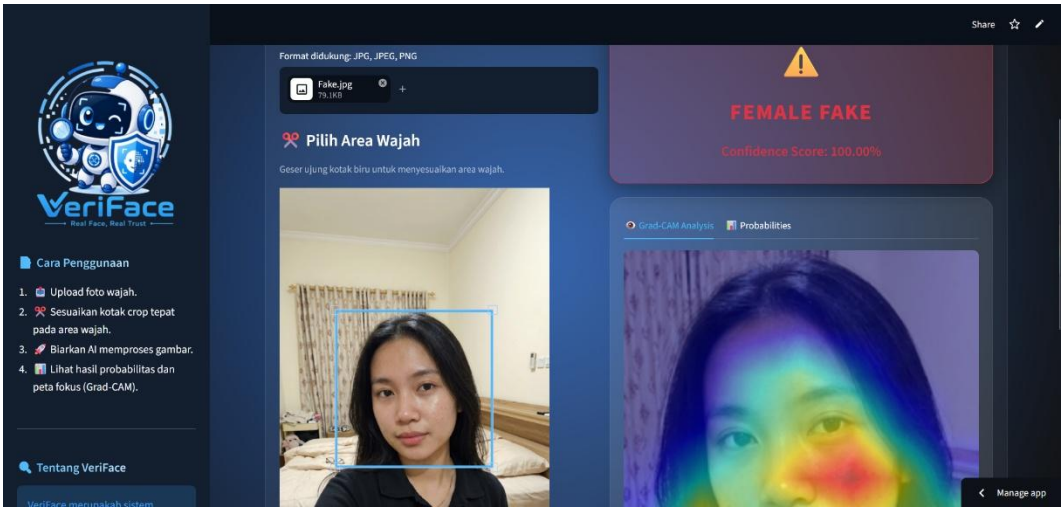
Bagian utama (*main content*) menampilkan komponen *upload* gambar yang mendukung format JPG, JPEG, dan PNG. Sebelum citra diunggah, halaman hanya

menampilkan area *upload* beserta informasi mengenai fungsi sistem. Pada kondisi ini belum dilakukan proses inferensi maupun visualisasi hasil prediksi karena aplikasi menunggu masukan berupa citra wajah dari pengguna. Pendekatan ini memberikan antarmuka yang bersih (*clean interface*) sehingga pengguna dapat langsung berfokus pada proses unggah citra sebagai langkah awal deteksi.

Selain menyediakan fasilitas *upload*, halaman awal juga dilengkapi dengan informasi mengenai pentingnya teknologi deteksi *deepfake*. Informasi tersebut menjelaskan secara ringkas tujuan pengembangan sistem, yaitu membantu membedakan citra wajah asli dan hasil manipulasi berbasis kecerdasan buatan dengan memanfaatkan karakteristik visual wajah orang Indonesia. Penyajian informasi ini diharapkan dapat memberikan pemahaman awal kepada pengguna mengenai fungsi dan manfaat aplikasi sebelum melakukan proses deteksi.

### 5.12.2. Tampilan *Website* Setelah Pengguna Mengunggah Gambar

Setelah pengguna mengunggah citra wajah, sistem secara otomatis menampilkan halaman analisis yang berisi proses prapemrosesan, hasil prediksi model, visualisasi Grad-CAM, serta distribusi probabilitas untuk setiap kelas. Tampilan *website* setelah pengguna mengunggah gambar ditunjukkan pada Gambar 5.25.




**Gambar 5.25.** Tampilan Hasil Analisis pada *Website*

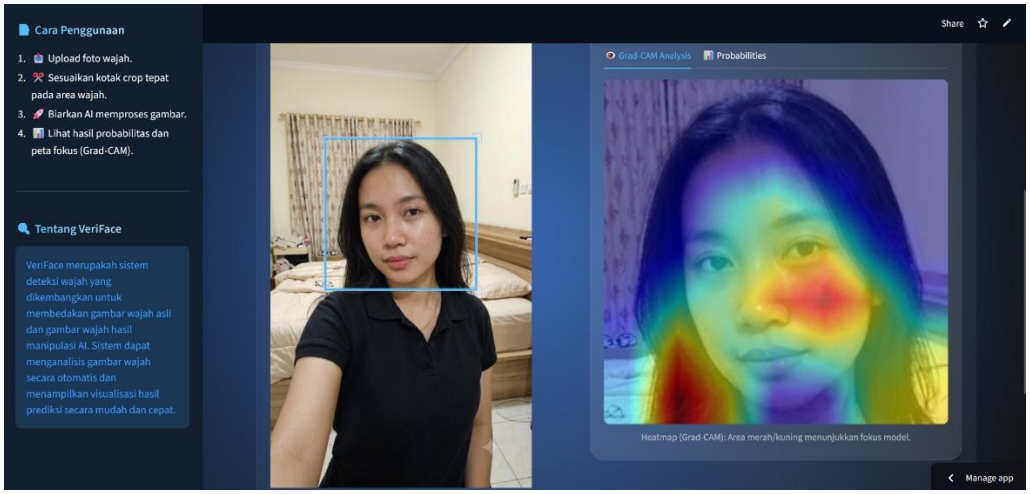
Berdasarkan tampilan *website* pada Gambar 5.25, setelah gambar berhasil diunggah, sistem menampilkan area "Pilih Area Wajah" yang memungkinkan pengguna melakukan proses *cropping* secara manual. Fitur ini disediakan agar area yang akan dianalisis benar-benar berfokus pada wajah sehingga informasi latar

belakang (*background*) yang tidak relevan dapat diminimalkan. Pengguna dapat menggeser maupun mengubah ukuran kotak *crop* sesuai dengan posisi wajah sebelum citra diproses oleh model. Setelah proses *cropping* selesai, citra secara otomatis diubah ukurannya menjadi 224 × 224 piksel sesuai dengan ukuran masukan yang digunakan selama proses pelatihan model EfficientNet-B0.

Selanjutnya, citra hasil *cropping* diproses melalui tahapan prapemrosesan yang meliputi konversi citra menjadi tensor serta normalisasi menggunakan nilai *mean* dan *standard deviation* yang sama dengan proses pelatihan model. Tahapan ini bertujuan untuk memastikan bahwa distribusi data masukan pada saat inferensi konsisten dengan data yang digunakan selama proses pelatihan sehingga model dapat memberikan prediksi yang optimal.

Pada penelitian ini, model menghasilkan empat kemungkinan kelas yaitu *Female Fake*, *Female Real*, *Male Fake*, dan *Male Real*. Berdasarkan contoh pengujian pada Gambar 5.25, sistem mengklasifikasikan citra sebagai *Female Fake* dengan nilai *confidence score* sebesar 100%. Hasil prediksi tersebut sesuai dengan label asli dari citra yang digunakan dalam pengujian, yaitu termasuk ke dalam kategori *Female Fake* atau wajah perempuan hasil manipulasi menggunakan teknologi *Artificial Intelligence* (*AI-generated*). Kesesuaian antara hasil prediksi dan label sebenarnya menunjukkan bahwa model EfficientNet-B0 mampu mengenali karakteristik visual yang membedakan wajah hasil sintesis AI dengan wajah asli pada sampel pengujian tersebut. Selain itu, nilai *confidence score* yang mencapai 100% menunjukkan bahwa model memiliki tingkat keyakinan yang sangat tinggi terhadap hasil klasifikasi yang diberikan.

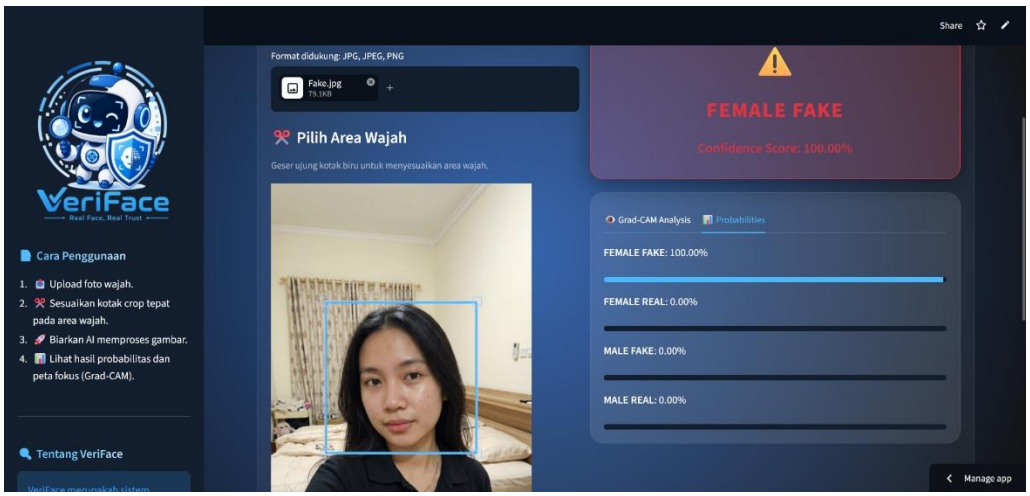
Selain menampilkan hasil prediksi, aplikasi juga menyediakan visualisasi Grad-CAM (*Gradient-weighted Class Activation Mapping*) pada *tab* Grad-CAM *Analysis*. Visualisasi ini berupa *heatmap* yang dioverlay pada citra wajah untuk menunjukkan area yang memperoleh perhatian terbesar dari model selama proses klasifikasi. Pada Gambar 5.26 terlihat bahwa area dengan warna merah hingga kuning berpusat pada bagian wajah, khususnya di sekitar hidung, pipi, mata, dan sebagian area mulut.




**Gambar 5.26.** Tampilan Hasil Grad-CAM

Mengingat citra yang diuji memang merupakan kategori *Female Fake*, fokus perhatian model pada area-area tersebut mengindikasikan bahwa keputusan klasifikasi dilakukan berdasarkan karakteristik visual yang dipelajari selama proses pelatihan, bukan secara acak. Dengan demikian, visualisasi Grad-CAM mendukung hasil prediksi model yang berhasil mengklasifikasikan citra sesuai dengan label sebenarnya. Sebaliknya, area yang berwarna biru menunjukkan bahwa bagian tersebut memberikan kontribusi yang relatif kecil terhadap keputusan model.

Selain visualisasi Grad-CAM, sistem juga menyediakan *tab Probabilities* yang menampilkan distribusi probabilitas untuk masing-masing kelas dalam bentuk persentase dan *progress bar*. Penyajian probabilitas ini memberikan informasi yang lebih komprehensif kepada pengguna mengenai tingkat keyakinan model terhadap seluruh kelas, tidak hanya kelas dengan probabilitas tertinggi.




**Gambar 5.27.** Tampilan Hasil Probabilitas Prediksi

Berdasarkan Gambar 5.27, model memberikan probabilitas sebesar 100% pada kelas *Female Fake*, sedangkan tiga kelas lainnya, yaitu *Female Real*, *Male Fake*, dan *Male Real*, masing-masing memperoleh probabilitas sebesar 0%. Distribusi probabilitas tersebut menunjukkan bahwa model memiliki tingkat keyakinan yang sangat tinggi terhadap hasil klasifikasi yang diberikan serta tidak menemukan karakteristik visual yang mengarah pada ketiga kelas lainnya. Selain meningkatkan transparansi hasil prediksi, informasi probabilitas juga membantu pengguna dalam mengevaluasi tingkat kepercayaan model terhadap setiap kemungkinan kelas, sehingga proses pengambilan keputusan menjadi lebih informatif dibandingkan hanya menampilkan label prediksi akhir.

Secara keseluruhan, halaman hasil analisis tidak hanya menyajikan keputusan klasifikasi berupa label dan *confidence score*, tetapi juga memberikan interpretasi visual melalui Grad-CAM serta distribusi probabilitas setiap kelas. Penyajian informasi tersebut meningkatkan transparansi sistem (*model interpretability*) karena pengguna dapat memahami dasar pengambilan keputusan model, bukan hanya menerima hasil prediksi akhir. Implementasi ini menjadikan *website* tidak hanya berfungsi sebagai alat klasifikasi wajah asli dan wajah *deepfake*, tetapi juga sebagai media yang mampu memberikan penjelasan terhadap proses inferensi yang dilakukan oleh model EfficientNet-B0.

# BAB VI
# PENUTUP

## 6.1. Kesimpulan

Berdasarkan seluruh tahapan penelitian yang meliputi pra-pemrosesan data, pelatihan model, serta evaluasi kinerja arsitektur EfficientNet-B0 dalam mendeteksi citra wajah asli dan *deepfake*, dapat dirumuskan beberapa kesimpulan utama sebagai berikut:

1. Evaluasi Konvergensi dan Kemampuan Generalisasi Model

   Arsitektur EfficientNet-B0 menunjukkan kemampuan dalam mengekstraksi fitur pada citra wajah. Model mencapai kondisi konvergensi dengan nilai akurasi validasi yang mencapai 100% pada beberapa *epoch*, disertai dengan penurunan nilai *loss* selama proses pelatihan. Keseimbangan antara kinerja pada data latih dan data validasi mengindikasikan bahwa model tidak mengalami *underfitting* maupun *overfitting*, sehingga memiliki kemampuan generalisasi yang baik.

2. Kinerja Model dan Akurasi

   Pada pengujian menggunakan data uji yang tidak digunakan selama pelatihan, model mencapai akurasi keseluruhan sebesar 98.00%. Model mampu membedakan empat kelas target (*female fake*, *female real*, *male fake*, dan *male real*) dengan kinerja yang konsisten, ditunjukkan oleh nilai rata-rata *F1-score* di atas 0.96 pada seluruh kelas. Selain itu, kurva ROC dan nilai AUC pada rentang 0.999 hingga 1.000 menunjukkan kemampuan model dalam memisahkan kelas dengan baik serta mengindikasikan bahwa prediksi yang dihasilkan tidak bersifat acak.

3. *Robustness* dan *Explainability*

   Model menunjukkan kinerja yang baik saat diuji pada data *in-the-wild* yang berada di luar distribusi *dataset*, dengan nilai probabilitas prediksi di atas 98%. Hasil ini didukung oleh visualisasi t-SNE yang memperlihatkan pemisahan klaster ruang fitur yang jelas. Selain itu, visualisasi Grad-CAM menunjukkan bahwa model memfokuskan perhatian pada area wajah yang relevan, seperti tekstur kulit pada bagian pipi, *T-zone*, dan garis rahang, serta

memberikan kontribusi yang relatif rendah pada elemen non-wajah seperti latar belakang dan pakaian.

4. Identifikasi Batasan dan Kelemahan Model

   Meskipun menunjukkan kinerja yang baik, analisis kesalahan model mengindikasikan adanya keterbatasan model pada kondisi visual tertentu, terutama pada kelas wajah perempuan. Empat kesalahan prediksi dari total 200 data uji berkaitan dengan tiga faktor utama, yaitu penggunaan riasan kosmetik yang memengaruhi tekstur kulit, kemiripan fitur gender akibat hasil rekayasa wajah yang halus (*androgynous features*), serta degradasi kualitas citra (misalnya gambar buram atau *soft focus*) yang memiliki kemiripan dengan artefak kompresi pada *deepfake*.

## 6.2. Saran

Meskipun arsitektur EfficientNet-B0 dalam penelitian ini menunjukkan kinerja deteksi *deepfake* yang tinggi dengan tingkat akurasi mencapai 98%, analisis kesalahan (*error analysis*) mengidentifikasi adanya keterbatasan pada kondisi visual tertentu. Oleh karena itu, untuk pengembangan penelitian selanjutnya, beberapa saran dapat diajukan sebagai berikut:

1. Peningkatan Variasi Data

   Berdasarkan hasil *error analysis*, model menunjukkan keterbatasan pada citra dengan karakteristik visual tertentu seperti penggunaan riasan kosmetik yang intens (*cosmetic illusion*) serta fitur wajah yang cenderung halus (*androgynous features*). Oleh karena itu, penelitian selanjutnya disarankan untuk memperkaya *dataset* dengan citra wajah asli (*real*) yang mencakup variasi riasan, kondisi pencahayaan studio (*flat lighting*), serta keragaman fitur fisik lintas gender. Pengayaan data ini bertujuan untuk meningkatkan kemampuan model dalam membedakan antara efek penghalusan akibat modifikasi fisik dan artefak manipulasi algoritmik pada *deepfake*.

2. Penambahan Variasi Atribut Wajah

   Selain meningkatkan variasi karakteristik visual, penelitian selanjutnya juga disarankan untuk menambahkan citra dengan berbagai atribut yang dikenakan di sekitar area wajah, seperti hijab, surban, masker, kacamata, topi, maupun aksesoris lainnya. Penambahan variasi atribut ini diharapkan dapat

meningkatkan kemampuan generalisasi model terhadap kondisi yang lebih beragam sehingga kinerjanya tetap baik pada berbagai karakteristik pengguna di dunia nyata.

3. Optimasi Augmentasi Citra

   Hasil analisis kesalahan menunjukkan bahwa model memiliki keterbatasan pada citra dengan kualitas rendah, seperti kondisi *soft focus* atau *blurry*. Oleh karena itu, penelitian selanjutnya disarankan untuk mengembangkan strategi augmentasi data dengan memasukkan simulasi degradasi kualitas citra. Teknik seperti Gaussian blur, penambahan noise, serta simulasi artefak kompresi JPEG dapat diterapkan pada data latih. Pendekatan ini bertujuan untuk meningkatkan kemampuan model dalam membedakan antara penurunan kualitas akibat faktor perangkat kamera dan artefak yang dihasilkan oleh manipulasi *deepfake*.

4. Eksplorasi Arsitektur Klasifikasi Lanjutan

   Hasil penelitian menunjukkan bahwa EfficientNet-B0 memberikan kinerja yang baik dalam tugas klasifikasi *deepfake*. Namun, seiring dengan perkembangan metode generasi *deepfake* yang semakin kompleks, penelitian selanjutnya disarankan untuk melakukan perbandingan dengan arsitektur yang lebih baru seperti *Vision Transformer* (ViT) atau EfficientNetV2. Pendekatan ini bertujuan untuk meninjau potensi mekanisme *self-attention* dalam menangkap pola distorsi yang lebih kompleks dibandingkan arsitektur berbasis konvolusi. Selain itu, penerapan metode *ensemble learning* yang menggabungkan beberapa model juga dapat dipertimbangkan untuk meningkatkan kinerja klasifikasi dan mengurangi kesalahan prediksi.

5. Pengembangan Deteksi *Deepfake* pada Modalitas Lain

   Penelitian selanjutnya dapat memperluas deteksi *deepfake* ke modalitas lain selain citra wajah, seperti suara (*deepfake voice*) dan video (*deepfake video*). Pada modalitas suara, penelitian dapat difokuskan pada deteksi suara asli dan hasil manipulasi serta klasifikasi emosi (*real emotion* dan *fake emotion*). Sementara itu, pada modalitas video, penelitian dapat difokuskan pada klasifikasi video asli dan video hasil manipulasi (*deepfake*). Pengembangan

ini diharapkan dapat memperluas penerapan teknologi deteksi *deepfake* pada berbagai jenis media digital.

# DAFTAR PUSTAKA


Adventino Gulo, S., Amelia Pertiwi, A., Putri Syaifullah Nasution, S., & Syahputra, H. (2025). Deteksi Deepfake Dalam Citra Menggunakan Convolutional Neural Network (Cnn). *JATI (Jurnal Mahasiswa Teknik Informatika)*, *9*(5), 8655–8660. https://doi.org/10.36040/jati.v9i5.14896

Agoncillo, R. A. F., Kiunisala, J. K. M., Raymund, M., & Cortez, D. M. A. (2024). *Enhancement of Deepfake Detection Framework Integrating Efficient NetB0 , Graph Attention Networks , and Gated Recurrent Units*. *5*(12), 16–19.

Alom, M. Z., Taha, T. M., Yakopcic, C., Westberg, S., Sidike, P., Nasrin, M. S., Hasan, M., Van Essen, B. C., Awwal, A. A. S., & Asari, V. K. (2019). A state-of-the-art survey on deep learning theory and architectures. *Electronics (Switzerland)*, *8*(3), 1–66. https://doi.org/10.3390/electronics8030292

Altuncu, E., Franqueira, V. N. L., & Li, S. (2024). Deepfake: definitions, performance metrics and standards, datasets, and a meta-review. *Frontiers in Big Data*, *7*. https://doi.org/10.3389/fdata.2024.1400024

Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M. A., Al-Amidie, M., & Farhan, L. (2021). Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. In *Journal of Big Data* (Vol. 8, Issue 1). https://doi.org/10.1186/s40537-021-00444-8

Bartos, G. E., Regia, A., Faculty, T., & Akyol, S. (2023). Deep Learning for Image Authentication : A Comparative Study on Real and AI-Generated Image Classification. *In Proceedings of the 18th International Symposium on Applied Informatics and Related Areas (AIS 2023)*.

Bishop, C. M. (2006). Pattern Recognition and Machine Learning. In *Springer* (Vol. 4, Issue 1). https://doi.org/10.53759/7669/jmc202404020

Cieslak, M. C., Castelfranco, A. M., Roncalli, V., Lenz, P. H., & Hartline, D. K. (2020). t-Distributed Stochastic Neighbor Embedding (t-SNE): A tool for eco-physiological transcriptomic analysis. *Marine Genomics*, *51*(September), 100723. https://doi.org/10.1016/j.margen.2019.100723

Dey, M. (2025). *Deepfake Statistics By Types, Fraud, Crime, Scams and Facts*

*(2026)*. ElectroIQ. https://electroiq.com/stats/deepfake-statistics/

Engelen, J. E. Van, & Hoos, H. H. (2020). A survey on semi-supervised learning. *Machine Learning*, *109*, 373–440. https://doi.org/https://doi.org/10.1007/s10994-019-05855-6

Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters*, *27*(8), 861–874. https://doi.org/10.1016/j.patrec.2005.10.010

Ghosh, M., & Thirugnanam, A. (2019). Introduction to Artificial Intelligence. In *Springer Nature Singapore*. https://www.researchgate.net/publication/351758474_Introduction_to_Artificial_Intelligence

Gimeno, P., Mingote, V., Ortega, A., Miguel, A., & Lleida, E. (2021). Generalizing AUC Optimization to Multiclass Classification for Audio Segmentation with Limited Training Data. *IEEE Signal Processing Letters*, *28*, 1135–1139. https://doi.org/10.1109/LSP.2021.3084501

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial nets. *Advances in Neural Information Processing Systems*, *3*(January), 2672–2680. https://doi.org/10.1007/978-3-658-40442-0_9

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2020). Generative adversarial networks. *Communications of the ACM*, *63*(11), 139–144. https://doi.org/10.1145/3422622

Goodfellow, I., Yoshua, B., & Courville, A. (2016). Deep learning. *Cambridge: MIT Press*, *19*(1–2), 305–307. https://doi.org/10.1007/s10710-017-9314-z

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *2016-Decem*, 770–778. https://doi.org/10.1109/CVPR.2016.90

Hijazi, S., Kumar, R., & Rowen, C. (2015). *Using Convolutional Neural Networks for Image Recognition*.

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

Hu, J., Shen, L., Albanie, S., Sun, G., & Wu, E. (2019). Squeeze-and-Excitation Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *42*(8), 2011–2023. https://doi.org/10.1109/TPAMI.2019.2913372

Khalil, M. (2025). *Deepfake Statistics 2025: AI Fraud Data & Trends*. DeepStrike. https://deepstrike.io/blog/deepfake-statistics-2025

Komdigi. (2025). *Hadapi Ancaman Deepfake, Komdigi Perkuat Literasi Digital dan Perlindungan Anak*. Komdigi – Portal Publik. https://portal.komdigi.go.id/kanal-publik/berita-kini/9498

Kotsiantis, S. B. (2007). Supervised machine learning: A review of classification techniques. *Informatica (Ljubljana)*, *31*(3), 249–268.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. *NeurIPS*, 1–9.

Kufel, J., Bargieł-Łączek, K., Kocot, S., Koźlik, M., Bartnikowska, W., Janik, M., Czogalik, Ł., Dudek, P., Magiera, M., Lis, A., Paszkiewicz, I., Nawrat, Z., Cebula, M., & Gruszczyńska, K. (2023). What Is Machine Learning, Artificial Neural Networks and Deep Learning?—Examples of Practical Applications in Medicine. *Diagnostics*, *13*(15). https://doi.org/10.3390/diagnostics13152582

Lecun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444. https://doi.org/10.1038/nature14539

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278–2323. https://doi.org/10.1109/5.726791

Mahmud, B. U., & Sharmin, A. (2021). *Deep Insights of Deepfake Technology : A Review*. http://arxiv.org/abs/2105.00192

Mahmud, K. H., Adiwijaya, & Faraby, S. Al. (2019). Klasifikasi Citra Multi-Kelas Menggunakan Convolutional Neural Network Studi Terkait Residual Neural Network. *E-Proceeding of Engineering*, *6*(1), 2127–2136.

Malik, A., Kuribayashi, M., Abdullahi, S. M., & Khan, A. N. (2022). DeepFake Detection for Human Face Images and Videos: A Survey. *IEEE Access*, *10*, 18757–18775. https://doi.org/10.1109/ACCESS.2022.3151186

Mathew, A., Amudha, P., & Sivakumari, S. (2021). Deep learning techniques: an overview. *Advances in Intelligent Systems and Computing*, *1141*, 599–608. https://doi.org/10.1007/978-981-15-3383-9_54

McCarthy, J. (2007). *What Is Artificial Intelligence?* Stanford University. http://www-formal.stanford.edu/jmc/whatisai/

Murdiawati, S., Wahyudin, A. R., Putra, J. A., & Suryono, R. R. (2026). Peran Machine Learning Dalam E-Commerce: Tinjauan Literatur Sistematis Terhadap Penerapan Dan Tantangan. *Jurnal Ilmiah ILKOMINFO (Ilmu Komputer & Informatika)*, *9*(2), 171–181.

Murphy, K. P. (2012). Machine Learning A Probabilistic Perspective. In *MIT Press*. https://doi.org/10.1007/978-94-011-3532-0_2

Nahm, F. S. (2022). Receiver operating characteristic curve: overview and practical use for clinicians. *Korean Journal of Anesthesiology*, *75*(1), 25–36. https://doi.org/10.4097/kja.21209

Nair, V., & Hinton, G. E. (2010). Recti fi ed Linear Units Improve Restricted Boltzmann Machines. *Proceedings of ICML*, *3*.

Nasteski, V. (2017). An overview of the supervised machine learning methods. *Horizons.B*, *4*, 51–62. https://doi.org/10.20544/horizons.b.04.1.17.p05

Nguyen, Q. H., Ly, H. B., Ho, L. S., Al-Ansari, N., Van Le, H., Tran, V. Q., Prakash, I., & Pham, B. T. (2021). Influence of data splitting on performance of machine learning models in prediction of shear strength of soil. *Mathematical Problems in Engineering*, *2021*. https://doi.org/10.1155/2021/4832864

Nugroho, P. A., Fenriana, I., & Arijanto, R. (2020). Implementasi Deep Learning Menggunakan Convolutional Neural Network (CNN) Pada Ekspresi Manusia. *Algor*, *2*(1), 12–21.

Nurfansepta, A. G., Muflikhah, L., & Setiawan, B. D. (2025). Deteksi Mutasi Epidermal Growth Factor Receptor pada Pasien Kanker Paru Menggunakan Algoritma Random Forest. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, *9*.

O'shea, K., & Nash, R. (2015). An Introduction to Convolutional Neural Networks. *International Journal for Research in Applied Science and Engineering*

*Technology*, *10*(12), 943–947. https://doi.org/10.22214/ijraset.2022.47789

Pahlawan, M. R., Setyanto, A., & Arief, M. R. (2024). Review Komprehensif Penggunaan Data Imbalance Pada Metode Klasifikasi Dalam Machine Learning. *Journal of Electrical Engineering and Computer (JEECOM)*, *xx*(xx). https://doi.org/10.33650/jeecom.v4i2

Pandey, D., Niwaria, K., & Chourasia, B. (2019). A review on machine learning algorithms. *International Research Journal of Engineering and Technology (IRJET)*, *06*(02), 920.

Patel, Y., Tanwar, S., Bhattacharya, P., Gupta, R., Alsuwian, T., Davidson, I. E., & Mazibuko, T. F. (2023). An Improved Dense CNN Architecture for Deepfake Image Detection. *IEEE Access*, *11*(March), 22081–22095. https://doi.org/10.1109/ACCESS.2023.3251417

Powers, D. M. W. (2011). *Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation*. 37–63. http://arxiv.org/abs/2010.16061

Pudjianto, B. W. (2025). *INDEKS MASYARAKAT DIGITAL INDONESIA 2025 INDONESIA DIGITAL SOCIETY INDEX 2025*.

Raihan, A. (2023). A comprehensive review of artificial intelligence and machine learning applications in the energy sector. *Journal of Technology Innovations and Energy*, *2*, 5. https://doi.org/https://doi.org/10.56556/jtie.v2i4.608

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., & Ommer, B. (2022). High-Resolution Image Synthesis with Latent Diffusion Models. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *2022-June*, 10674–10685. https://doi.org/10.1109/CVPR52688.2022.01042

Roose, K. (2022, September 2). An A.I.-Generated Picture Won an Art Prize. Artists Aren't Happy. *The New York Times*. https://www.nytimes.com/2022/09/02/technology/ai-artificial-intelligence-artists.html

Rossler, A., Cozzolino, D., Verdoliva, L., Riess, C., Thies, J., & Niessner, M. (2019). FaceForensics++: Learning to detect manipulated facial images. *Proceedings of the IEEE International Conference on Computer Vision*, 1–11.

https://doi.org/10.1109/ICCV.2019.00009

Russell, S. J., & Norvig, P. (2021). *Artificial Intelligence A Modern Approach Fourth Edition*. Pearson Education.

Şafak, E., & Barışçı, N. (2024). Detection of fake face images using lightweight convolutional neural networks with stacking ensemble learning method. *PeerJ Computer Science*, *10*, 1–20. https://doi.org/10.7717/PEERJ-CS.2103

Safitri, I. K. (2025). *Deepfake, tipuan canggih dunia maya*. Tempo – Interaktif. https://interaktif.tempo.co/proyek/deepfake-tipuan-canggih-dunia-maya/index.html

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018). MobileNetV2: Inverted Residuals and Linear Bottlenecks. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 4510–4520. https://doi.org/10.1109/CVPR.2018.00474

Satrio, F. A. (2025). *Mafindo: Deepfake dan Penipuan Digital Jadi Ancaman Serius Tahun 2025*. Jakarta Times. https://jakarta.times.co.id/news/berita/E0U7z3q7D/Mafindo-Deepfake-dan-Penipuan-Digital-Jadi-Ancaman-Serius-Tahun-2025

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2017). Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. *International Journal of Computer Vision*, *128*(2), 336–359. https://doi.org/10.1007/s11263-019-01228-7

Shad, H. S., Rizvee, M. M., Roza, N. T., Hoq, S. M. A., Monirujjaman Khan, M., Singh, A., Zaguia, A., & Bourouis, S. (2021). Retraction:Comparative Analysis of Deepfake Image Detection Method Using Convolutional Neural Network. *Computational Intelligence and Neuroscience*, *2021*. https://doi.org/10.1155/2021/3111676

Shahzad, H. F., Rustam, F., Flores, E. S., Luís Vidal Mazón, J., de la Torre Diez, I., & Ashraf, I. (2022). A Review of Image Processing Techniques for Deepfakes. *Sensors*, *22*(12), 1–28. https://doi.org/10.3390/s22124556

Simeone, O. (2018). A brief introduction to machine learning for engineers. *Foundations and Trends in Signal Processing*, *12*(3–4), 200–431. https://doi.org/10.1561/2000000102

Singh, H. (2017). Artificial Intelligence Revolution and India's AI Development: Challenges and Scope. *International Journal of Scientific Research in Science, Engineering and Technology*, *3*(3), 417–421. https://ijsrset.com/IJSRSET1733103

Sivakumar, M., Parthasarathy, S., & Padmapriya, T. (2024). Trade-off between training and testing ratio in machine learning for medical image processing. *PeerJ Computer Science*, *10*, 1–17. https://doi.org/10.7717/PEERJ-CS.2245

Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing and Management*, *45*(4), 427–437. https://doi.org/10.1016/j.ipm.2009.03.002

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2015). Going deeper with convolutions. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *07-12-June*, 1–9. https://doi.org/10.1109/CVPR.2015.7298594

Tan, M., & Le, Q. V. (2019). EfficientNet: Rethinking model scaling for convolutional neural networks. *36th International Conference on Machine Learning, ICML 2019*, *2019-June*, 10691–10700.

Tolosana, R., Vera-Rodriguez, R., Fierrez, J., Morales, A., & Ortega-Garcia, J. (2020). Deepfakes and beyond: A Survey of face manipulation and fake detection. *Information Fusion*, *64*(9), 131–148. https://doi.org/10.1016/j.inffus.2020.06.014

Tripathi, S. M. (2025). *Classification of AI-Generated , Deepfake and Real Images Using CNN*. *12*(6), 555–560.

Unit Survey APJII. (2025). Survei Penetrasi Internet dan Perilaku Penggunaan Internet Sebagai perwakilan Pengurus APJII, kami dengan bangga mempersembahkan Profil Internet Indonesia 2025. In *Asosiasi Penyelenggara Jasa Internet Indonesia*. https://survei.apjii.or.id/

Verdoliva, L. (2020). Media Forensics and DeepFakes: An Overview. *IEEE Journal on Selected Topics in Signal Processing*, *14*(5), 910–932. https://doi.org/10.1109/JSTSP.2020.3002101

Wang, T., Lu, C., Yang, M., Hong, F., & Liu, C. (2020). A hybrid method for

heartbeat classification via convolutional neural networks, multilayer perceptrons and focal loss. *PeerJ Computer Science*, *6*(Cvd), 3–17. https://doi.org/10.7717/PEERJ-CS.324

Westerlund, M. (2019). The emergence of deepfake technology: A review. *Technology Innovation Management Review*, *9*(11), 39–52. https://doi.org/10.22215/TIMREVIEW/1282

Yan, J., & Wang, X. (2022). Unsupervised and semi-supervised learning: the next frontier in machine learning for plant systems biology. *Plant Journal*, *111*(6), 1527–1538. https://doi.org/10.1111/tpj.15905

Yoga Wibowo, M., Hikmayanti, H., Fitri Nur Masruriyah, A., Novalia, E., & Heryana, N. (2023). Mask Use Detection in Public Places Using the Convolutional Neural Network Algorithm. *Edutran Computer Science and Information Technology*, *1*(1), 19–25. https://doi.org/10.59805/ecsit.v1i1.10

# LAMPIRAN

Lampiran 1 *Link* Google Drive *Syntax*

https://drive.google.com/drive/folders/1njB23XbQnRwbg-CxTo6-Gtv7yNCGEr9D?usp=sharing