

Terbit online pada laman : <http://teknosi.fti.unand.ac.id/>

## Jurnal Nasional Teknologi dan Sistem Informasi

| ISSN (Print) 2460-3465 | ISSN (Online) 2476-8812 |



Artikel Penelitian

# Implementasi Transfer Learning Menggunakan DenseNet121 untuk Deteksi Presentation Attack pada Citra Wajah

Septian Nuno Zildjian <sup>a,\*</sup>, Dian Nurdiana <sup>b</sup>, Fonda Leviany <sup>c</sup>

<sup>a,b,c</sup> Program Studi Sains Data, Fakultas Sains & Teknologi, Universitas Terbuka, Jln Cabe Raya, Pondok Cabe, Pamulang, Tangerang Selatan, Banten 15437, Indonesia

### INFORMASI ARTIKEL

#### Sejarah Artikel:

Diterima Redaksi: 03 Juli 2026

Revisi Akhir: 20 Agustus 2026

Diterbitkan Online: 24 Agustus 2026

### KATA KUNCI

Presentation Attack Detection,

DenseNet121,

Face Anti-Spoofing,

Transfer Learning,

Image Classification

### KORESPONDENSI

E-mail: [septiannunozildjian@gmail.com](mailto:septiannunozildjian@gmail.com)\*

### A B S T R A C T

Ujian jarak jauh menjadi tulang punggung penyelenggaraan Pendidikan Jarak Jauh (PJJ) di Indonesia, terutama pada institusi berskala masif seperti Universitas Terbuka dan platform Massive Open Online Courses (MOOC), namun keandalan verifikasi wajah pada sistem proctoring komersial terus diuji oleh ancaman *presentation attack* atau spoofing wajah. Dengan bermadankan foto cetak, foto pada layar perangkat lain, masker, atau manekin, joki ujian dapat mengelabui verifikasi wajah awal tanpa keahlian teknis khusus dan biaya rendah, kemudian bebas mengerjakan ujian hingga selesai tanpa terdeteksi. Skala pengawasan manual pada ratusan ribu peserta ujian PJJ membuat verifikasi satu per satu menjadi tidak mungkin dilakukan, sehingga dibutuhkan solusi deteksi otomatis yang cepat, ringan, dan andal untuk memfilter tangkapan webcam secara masal pada infrastruktur server kampus. Penelitian ini menjawab kebutuhan tersebut dengan mengembangkan model deteksi *presentation attack* menggunakan pendekatan *transfer learning* pada arsitektur DenseNet121, dilatih pada dataset citra wajah multikelas berjumlah 1.403 citra yang mencakup enam kategori: *fake\_mannequin*, *fake\_mask*, *fake\_printed*, *fake\_screen*, *fake\_unknown*, dan *realperson*. Model dilatih menggunakan strategi *fine-tuning* parsial pada blok konvolusi akhir (*denseblock4* dan *norm5*) dengan optimizer AdamW, penjadwal *ReduceLROnPlateau*, dan *early stopping* untuk menjaga performa sekaligus mencegah *overfitting*. Hasil pengujian menunjukkan akurasi 87,68%, *precision* tertimbang 88,79%, *recall* tertimbang 87,68%, dan *F1-score* tertimbang 87,62%, dengan performa stabil dan konsisten pada seluruh kelas serangan. Model terbaik diimplementasikan ke aplikasi antarmuka berbasis Streamlit untuk mendemonstrasikan deteksi secara interaktif dan mudah dipahami pengguna non-teknis. Hasil ini membuktikan bahwa pendekatan *transfer learning* berbasis DenseNet121 akurat, ringan, dan efisien untuk diadopsi sebagai lapisan pertahanan tambahan pada sistem proctoring ujian daring di lingkungan Pendidikan Jarak Jauh berskala besar, di tengah meningkatnya ancaman spoofing terhadap integritas ujian.

## 1. PENDAHULUAN

Verifikasi wajah kini menjadi salah satu metode autentikasi biometrik yang paling banyak digunakan untuk memastikan keaslian identitas seseorang secara jarak jauh tanpa memerlukan kehadiran fisik. Berbagai sektor, mulai dari perbankan digital, layanan pemerintahan, hingga pendidikan jarak jauh, kini mengandalkan verifikasi berbasis citra wajah sebagai mekanisme utama otentikasi pengguna [1], [2], karena pendekatan ini bersifat

noninvasif, tidak memerlukan perangkat keras tambahan, dan mudah diintegrasikan ke antarmuka digital yang sudah ada [3].

Tantangan ini terlihat nyata pada penyelenggaraan ujian daring di perguruan tinggi, di mana banyak institusi mengandalkan aplikasi *proctoring* pihak ketiga seperti Safe Exam Browser, Respondus, atau Examudo untuk mengawasi peserta ujian dari jarak jauh, sebuah adopsi teknologi pengawasan yang meluas pesat namun juga menuai kritik terkait penyerahan kendali kepada penyedia komersial [4]. Kelemahan utama sistem semacam ini terletak pada mekanisme verifikasi wajah yang umumnya hanya

dilakukan sekali di awal sesi ujian: begitu tahap tersebut berhasil dikelabui menggunakan foto cetak (*fake printed*) atau medium presentation attack lainnya, joki dapat mengerjakan ujian hingga selesai tanpa terdeteksi oleh sistem [5]. Kondisi ini menunjukkan bahwa keberadaan fitur verifikasi wajah semata tidak cukup tanpa disertai kemampuan deteksi presentation attack yang berjalan konsisten sepanjang sesi ujian, bukan hanya pada tahap awal.

Urgensi ini semakin terasa pada konteks Pendidikan Jarak Jauh (PJJ) berskala masif, seperti penyelenggaraan *Massive Open Online Courses* (MOOC) maupun universitas terbuka. Sebagai pelopor PJJ di Indonesia, Universitas Terbuka mengelola ratusan ribu mahasiswa yang melaksanakan ujian akhir semester secara daring dari rumah masing-masing. Pada skala populasi peserta sebesar ini, pengawasan manual oleh sejumlah kecil staf pengawas terhadap setiap sesi verifikasi wajah menjadi tidak mungkin (*impossible*) dan rawan kesalahan untuk dilakukan secara konsisten [6]. Di sinilah urgensi AI pendeteksi presentation attack sangat dibutuhkan: model deteksi yang ringan dan otomatis memungkinkan puluhan ribu tangkapan webcam disaring secara cepat dan murah pada infrastruktur server kampus, tanpa bergantung pada kapasitas pengawas manusia yang terbatas [6]. Kejadian dan kondisi ini menegaskan bahwa ancaman presentation attack bukan sekadar isu teoretis, melainkan tantangan nyata yang harus dijawab agar integritas ujian pada penyelenggaraan PJJ berskala besar tetap terjaga.

Presentation attack sendiri merupakan upaya manipulasi yang menyajikan representasi wajah palsu kepada sensor kamera untuk mengelabui autentikasi, melalui medium seperti foto cetak, foto pada layar perangkat lain, masker wajah tiga dimensi, hingga manekin [7], [8]. Serangan ini umumnya berbiaya rendah dan tidak memerlukan keahlian teknis tinggi sehingga rawan dieksploitasi secara luas [9], sementara sistem verifikasi wajah konvensional tanpa mekanisme deteksi yang memadai tidak dapat menjamin bahwa wajah yang tertangkap kamera benar-benar berasal dari individu yang berhak [9], [10]. Urgensi ini semakin meningkat seiring masifnya adopsi autentikasi biometrik wajah pada layanan berskala besar, di mana verifikasi manual satu per satu tidak lagi praktis untuk volume pengguna yang bisa mencapai ribuan hingga ratusan ribu permintaan dalam satu periode [1], sehingga dibutuhkan lapisan deteksi otomatis yang cepat, konsisten, dan tetap ringan untuk dijalankan pada infrastruktur umum.

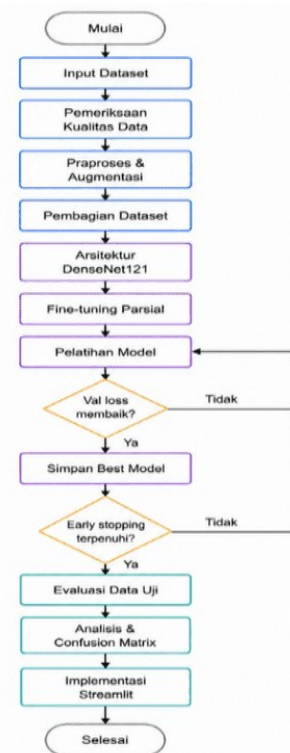
Pendekatan generasi awal seperti deteksi kedipan mata (*eyeblick*) dan analisis tekstur warna (*color texture analysis*) terbukti kurang robust terhadap variasi pencahayaan, jenis kamera, dan modalitas serangan baru [7], [9], [10]. Perkembangan *deep learning*, khususnya convolutional neural network (CNN), kemudian mendorong pergeseran menuju model yang mampu mempelajari representasi fitur tekstur, pencahayaan, dan artefak visual secara otomatis tanpa rekayasa fitur manual [11], [12]. Beberapa penelitian telah menerapkan *transfer learning* dengan arsitektur CNN pratatih untuk deteksi presentation attack, di antaranya MobileNetV2 [12], ResNet50 [13], dan VGG16 [14], serta DenseNet dengan mekanisme perhatian kanal (*channel attention*) [15]. Namun, eksplorasi DenseNet121 dengan strategi *fine-tuning* parsial pembekuan sebagian blok konvolusi awal dan pelatihan ulang blok akhir untuk kasus presentation attack multikelas masih terbatas [16]. Kesenjangan inilah yang menjadi dasar penelitian

ini: mengeksplorasi potensi DenseNet121 [16], yang unggul dalam efisiensi parameter dan aliran gradien melalui mekanisme dense connection antarlapis, sehingga potensial diterapkan pada dataset berukuran terbatas seperti yang umum dijumpai dalam konteks penelitian lokal.

Penelitian ini memanfaatkan *transfer learning* dengan DenseNet121 [16] yang telah dilatih pada ImageNet [17], kemudian di-*fine-tuning* pada dataset citra wajah multikelas mencakup lima kategori serangan presentasi dan satu kategori wajah asli. Kontribusi penelitian ini meliputi: (1) penyusunan pipeline klasifikasi presentation attack enam kelas berbasis DenseNet121 dengan *fine-tuning* parsial; (2) evaluasi performa menyeluruh menggunakan metrik akurasi, precision, recall, F1-score, confusion matrix, dan analisis kesalahan per kelas; serta (3) implementasi model ke dalam aplikasi antarmuka berbasis Streamlit sebagai prototipe modul deteksi presentation attack pada sistem autentikasi biometrik wajah.

## 2. METODE

Penelitian ini menggunakan pendekatan eksperimental berbasis *deep learning* dengan arsitektur DenseNet121 melalui strategi *transfer learning* dan *fine-tuning* parsial. Secara keseluruhan, alur penelitian terdiri atas tahap persiapan data, pembangunan dan pelatihan model, evaluasi performa, hingga implementasi prototipe antarmuka deteksi. Gambar 1 berikut menyajikan diagram alir keseluruhan tahapan yang dilakukan.



Gambar 1. Diagram Alir Penelitian

### 2.1. Penelitian Sebelumnya

Penelitian mengenai deteksi presentation attack pada sistem verifikasi wajah telah berkembang dari pendekatan berbasis fitur

tangan (*hand-crafted feature*) menuju pendekatan berbasis *deep learning*. Pada generasi awal, Boulkenafet et al. mengusulkan analisis tekstur warna (*color texture analysis*) yang memanfaatkan perbedaan pola refleksi cahaya antara kulit asli dan medium pemalsuan [7], namun pendekatan semacam ini umumnya kurang robust terhadap variasi pencahayaan, jenis kamera, dan modalitas serangan baru yang tidak terlihat pada tahap pelatihan [10].

Perkembangan *deep learning* kemudian mendorong pergeseran pendekatan menuju *convolutional neural network* (CNN) yang mampu mempelajari representasi fitur secara otomatis. Liu et al. mengusulkan kerangka pembelajaran mendalam dengan supervisi tambahan (*auxiliary supervision*) berupa estimasi peta kedalaman (*depth map*) untuk meningkatkan generalisasi model terhadap serangan yang belum pernah dijumpai [11]. Pada konteks penelitian lokal, beberapa studi telah menerapkan strategi *transfer learning* menggunakan arsitektur CNN pratatih, di antaranya Hafidz et al. yang menggunakan MobileNetV2 untuk deteksi serangan presentasi pada sistem verifikasi wajah [12], Rachmadi et al. yang menerapkan ResNet50 untuk sistem deteksi wajah palsu pada autentikasi biometrik [13], dan Nurhakim et al. yang mengombinasikan augmentasi data dengan *transfer learning* VGG16 untuk deteksi spoofing wajah [14]. Studi yang paling dekat dengan topik penelitian ini menggunakan arsitektur DenseNet dengan mekanisme perhatian kanal (*channel attention*) untuk deteksi kecurangan berbasis pengenalan wajah pada sistem verifikasi otomatis [15]. Ringkasan posisi masing-masing penelitian dibandingkan dengan penelitian ini disajikan pada Tabel 1

Tabel 1. Ringkasan Penelitian Sebelumnya terkait Presentation Attack

| N o | Peneliti (Ref.)        | Metode Arsitektur                                       | Konteks Penelitian                                | Hasil Utama                                                | Kesenjangan terhadap Penelitian Ini                                        |
|-----|------------------------|---------------------------------------------------------|---------------------------------------------------|------------------------------------------------------------|----------------------------------------------------------------------------|
| 1   | Boulkenafet et al. [7] | Analisis tekstur warna ( <i>color texture</i> )         | Face anti-spoofing berbasis fitur tangan          | Mendeteksi perbedaan refleksi cahaya kulit vs medium palsu | Fitur <i>hand-crafted</i> , rentan terhadap variasi pencahayaan & kamera   |
| 2   | Liu et al. [11]        | CNN + <i>auxiliary supervision</i> ( <i>depth map</i> ) | Face anti-spoofing berbasis <i>deep learning</i>  | Meningkatkan generalisasi terhadap serangan baru           | Memerlukan label tambahan ( <i>depth map</i> ), kompleks                   |
| 3   | Hafidz et al. [12]     | <i>Transfer learning</i> MobileNetV2                    | Deteksi serangan presentasi pada verifikasi wajah | Akurasi kompetitif dengan model ringan                     | Arsitektur lebih dangkal, kedalaman fitur terbatas dibanding DenseNet121   |
| 4   | Rachmadi et al. [13]   | <i>Transfer learning</i> ResNet50                       | Deteksi wajah palsu untuk autentikasi biometrik   | Akurasi baik pada deteksi wajah palsu                      | Tidak mengeksplorasi strategi <i>fine-tuning</i> parsial                   |
| 5   | Nurhakim et al. [14]   | <i>Transfer learning</i> VGG16 + augmentasi data        | Deteksi spoofing wajah                            | Augmentasi data meningkatkan akurasi                       | VGG16 memiliki parameter lebih besar, kurang efisien dibanding DenseNet121 |

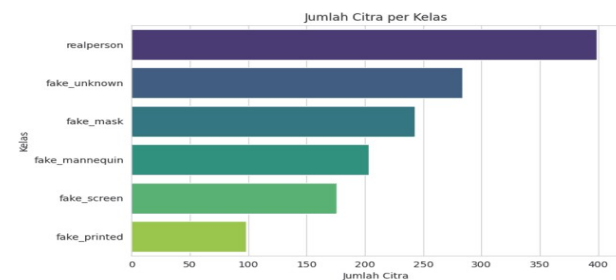
Berdasarkan Tabel 1, dapat dilihat bahwa penelitian-penelitian sebelumnya umumnya berfokus pada satu strategi tunggal baik penggunaan arsitektur CNN alternatif tanpa eksplorasi *fine-tuning* parsial [12], [13], [14], maupun penambahan modul *attention* di atas *backbone* DenseNet secara utuh [15] sedangkan strategi *fine-tuning* parsial pada DenseNet121, khususnya pembekuan sebagian blok konvolusi awal dan pelatihan ulang blok akhir untuk kasus *presentation attack* multikelas, masih terbatas dilakukan. Kesenjangan (*gap*) inilah yang menjadi dasar penelitian ini.

## 2.2. Dataset

Dataset yang digunakan dalam penelitian ini merupakan hasil kompilasi dari beberapa dataset citra wajah yang tersedia secara publik di platform Kaggle. Setiap dataset sumber dipilih berdasarkan kesesuaian jenis *presentation attack* yang direpresentasikan dengan enam kelas target penelitian, kemudian citra-citra dari berbagai sumber tersebut diseleksi, disaring ulang labelnya, dan digabungkan menjadi satu dataset yang homogen. Melalui proses kurasi ini, terbentuk dataset akhir yang terdiri atas 1.403 citra wajah yang terbagi ke dalam enam kelas, yaitu *fake\_mannequin*, *fake\_mask*, *fake\_printed*, *fake\_screen*, *fake\_unknown*, dan *realperson*. Distribusi jumlah citra per kelas ditunjukkan pada Tabel 2 dan Gambar 2.

Tabel 2. Distribusi Jumlah Citra per Kelas

| Kelas                 | Jumlah Citra | Persentase |
|-----------------------|--------------|------------|
| <i>fake_mannequin</i> | 203          | 14,47%     |
| <i>fake_mask</i>      | 243          | 17,32%     |
| <i>fake_printed</i>   | 98           | 6,99%      |
| <i>fake_screen</i>    | 176          | 12,55%     |
| <i>fake_unknown</i>   | 284          | 20,24%     |
| <i>realperson</i>     | 399          | 28,44%     |



Gambar 2. Distribusi jumlah citra per kelas pada dataset.

Berdasarkan Tabel 2 dan Gambar 2, terlihat bahwa distribusi kelas tidak seimbang, dengan kelas *realperson* sebagai kelas mayoritas (399 citra) dan *fake\_printed* sebagai kelas minoritas (98 citra), menghasilkan rasio ketidakseimbangan sebesar 4,07. Hasil pemeriksaan kualitas data menunjukkan tidak ditemukan citra yang rusak (*corrupt*), namun terdapat 74 citra duplikat berdasarkan hash MD5 yang tetap dipertahankan dalam dataset karena pemeriksaan duplikasi pada penelitian ini bersifat informatif dan tidak diikuti dengan proses penghapusan otomatis.



Gambar 3. Contoh citra representatif untuk setiap kelas pada dataset

Rata-rata resolusi citra pada dataset adalah  $998,1 \times 1.478,0$  piksel, sehingga seluruh citra perlu diseragamkan ukurannya pada tahap praproses sebelum dimasukkan ke dalam model.

### 2.3 Praproses dan Augmentasi Data

Setiap citra diubah ukurannya menjadi  $224 \times 224$  piksel agar sesuai dengan ukuran input standar DenseNet121 [16], kemudian dikonversi ke ruang warna RGB dan dinormalisasi menggunakan statistik *mean* dan standar deviasi ImageNet ([0,485; 0,456; 0,406] dan [0,229; 0,224; 0,225]) [17]. Pada data latih, diterapkan augmentasi untuk meningkatkan generalisasi model dan mensimulasikan variasi kondisi pengambilan citra pada skenario *presentation attack* di lapangan [14], meliputi penyesuaian acak kecerahan dan kontras (*ColorJitter*, rentang 0,3) serta kompresi JPEG acak dengan kualitas 40 pada 20% sampel untuk mensimulasikan artefak kompresi akibat pengambilan ulang dari layar atau kamera kedua [7]. Contoh hasil augmentasi disajikan pada Gambar 4.



Gambar 4. Perbandingan citra asli dengan hasil augmentasi

### 2.4. Pembagian Dataset

Dataset dibagi menjadi data Train, Validation, dan test dengan rasio 70:15:15 menggunakan stratified split berdasarkan label kelas agar proporsi setiap kelas tetap terjaga pada masing-masing subset. Hasil pembagian dataset diperoleh 982 citra data latih, 210 citra data validasi, dan 211 citra data Test, dengan rincian per kelas pada Tabel 3.

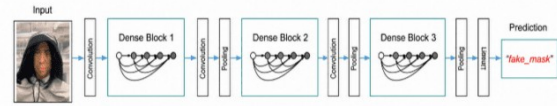
Tabel 3. Distribusi Data Pembagian Dataset

| Kelas          | Train | Validation | Test |
|----------------|-------|------------|------|
| fake_mannequin | 142   | 30         | 31   |
| fake_mask      | 170   | 36         | 37   |
| fake_printed   | 69    | 15         | 14   |
| fake_screen    | 123   | 27         | 26   |
| fake_unknown   | 199   | 42         | 43   |
| realperson     | 279   | 60         | 60   |
| Total          | 982   | 210        | 211  |

### 2.5. Arsitektur Model

DenseNet121 (*Densely Connected Convolutional Networks*) merupakan salah satu varian arsitektur jaringan saraf konvolusi yang diperkenalkan oleh Huang et al. [16], di mana setiap lapisan menerima masukan dari seluruh lapisan sebelumnya melalui koneksi padat (*dense connection*) dalam satu blok. Secara umum, arsitektur DenseNet terdiri atas beberapa *dense block* yang dihubungkan oleh *transition layer* berupa konvolusi dan *pooling*

untuk mengubah ukuran *feature map*. Konsep dasar arsitektur DenseNet tersebut diilustrasikan pada Gambar 5. Pendekatan ini memungkinkan aliran gradien yang lebih baik selama pelatihan, mendorong penggunaan ulang fitur antarlapis, serta memperkecil jumlah parameter dibandingkan arsitektur konvensional dengan kedalaman serupa [16].



Gambar 5. Ilustrasi konsep arsitektur DenseNet dengan tiga *dense block & transition layer*.

Model deteksi *presentation attack* dibangun berdasarkan arsitektur DenseNet121 yang telah dilatih sebelumnya pada dataset ImageNet (bobot IMAGENET1K\_V1) [17]. Strategi *transfer learning* yang digunakan adalah *fine-tuning* parsial, yaitu membekukan (*freeze*) seluruh lapisan ekstraksi fitur pada tahap awal, kemudian membuka kembali parameter pada blok *denseblock4* dan lapisan *norm5* agar ikut diperbarui selama pelatihan, sementara blok-blok konvolusi awal tetap dibekukan. Lapisan *classifier* bawaan diganti dengan blok *Dropout* (rate 0,3) yang diikuti lapisan *Linear* berukuran enam unit keluaran, sesuai dengan jumlah kelas pada permasalahan klasifikasi ini [18].

Strategi ini menghasilkan model dengan total 6.960.006 parameter, dengan 2.166.278 parameter (31,12%) yang bersifat *trainable*. Pendekatan *fine-tuning* parsial pada blok akhir dipilih dengan pertimbangan bahwa fitur tingkat rendah dan menengah dari DenseNet121 hasil pra-pelatihan pada ImageNet sudah cukup representatif untuk citra wajah secara umum, sehingga proses pelatihan dapat difokuskan pada fitur tingkat tinggi yang lebih spesifik untuk membedakan karakteristik tekstur dan artefak pada *presentation attack*, sekaligus menjaga waktu pelatihan dan mengurangi risiko *overfitting* pada ukuran dataset yang relatif terbatas [19].

### 2.6. Konfigurasi Pelatihan

Proses pelatihan menggunakan fungsi loss CrossEntropyLoss dengan optimizer AdamW [20] yang hanya memperbarui parameter *trainable*, *learning rate* awal  $1 \times 10^{-4}$ , dan *weight decay*  $1 \times 10^{-4}$ . Penjadwal laju pembelajaran ReduceLROnPlateau diterapkan untuk menurunkan *learning rate* sebesar faktor 0,5 apabila *val\_loss* tidak membaik dalam 3 *epoch* berturut-turut [21]. Mekanisme *early stopping* dengan *patience* 7 *epoch* dan ambang minimal perubahan  $1 \times 10^{-4}$  digunakan untuk menghentikan pelatihan secara otomatis ketika *val\_loss* tidak lagi menunjukkan perbaikan signifikan, guna mencegah *overfitting* [18]. Ringkasan konfigurasi hyperparameter pelatihan disajikan pada Tabel 4.

Tabel 4. Konfigurasi Parameter

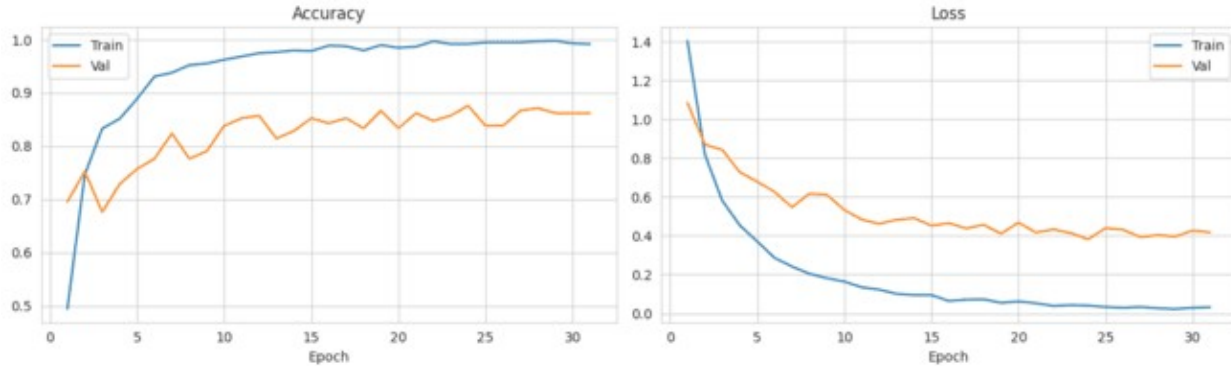
| Parameter             | Nilai              |
|-----------------------|--------------------|
| Ukuran citra input    | $224 \times 224$   |
| Ukuran <i>batch</i>   | 32                 |
| Maksimum <i>epoch</i> | 50                 |
| <i>Learning rate</i>  | $1 \times 10^{-4}$ |
| <i>Weight decay</i>   | $1 \times 10^{-4}$ |
| Optimizer             | AdamW              |

|                                |                   |
|--------------------------------|-------------------|
| <i>Scheduler</i>               | ReduceLROnPlateau |
| <i>Early stopping patience</i> | 7 epoch           |
| <i>Random seed</i>             | 42                |

### 3. HASIL & PEMBAHASAN

#### 3.1. Proses Pelatihan Model

Pelatihan berjalan selama 31 *epoch* sebelum dihentikan secara otomatis oleh mekanisme *early stopping*, dengan total waktu



Gambar 6. Kurva akurasi (kiri) dan loss (kanan) pada data latih dan data validasi selama proses pelatihan

Selisih antara akurasi data latih (di atas 99%) dan akurasi data validasi (sekitar 86–88%) pada *epoch-epoch* akhir mengindikasikan adanya kecenderungan *overfitting* ringan, yang merupakan konsekuensi umum dari ukuran dataset yang relatif terbatas (1.403 citra) untuk model dengan kapasitas parameter cukup besar [18].

Penerapan *dropout* [18], *weight decay* [20], augmentasi data, serta *early stopping* pada penelitian ini berperan dalam membatasi dampak *overfitting* tersebut sehingga performa pada data validasi dan data uji tetap terjaga pada rentang yang konsisten [22].

#### 3.2 Evaluasi pada Data Uji

Evaluasi akhir dilakukan menggunakan model dengan bobot terbaik (*best\_model*) pada 211 citra data uji yang belum pernah dilihat oleh model selama pelatihan maupun validasi. Performa model diukur menggunakan empat metrik evaluasi standar, yaitu akurasi, *precision*, *recall*, dan F1-score, yang dihitung berdasarkan jumlah prediksi benar positif (*true positive*, TP), benar negatif (*true negative*, TN), salah positif (*false positive*, FP), dan salah negatif (*false negative*, FN) untuk setiap kelas [Persamaan 1-4].

Akurasi mengukur proporsi prediksi yang benar terhadap keseluruhan data uji:

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

*Precision* mengukur proporsi prediksi positif suatu kelas yang benar-benar merupakan anggota kelas tersebut:

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

*Recall* mengukur proporsi anggota suatu kelas yang berhasil diprediksi dengan benar oleh model:

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

F1-score merupakan rata-rata harmonik antara *precision* dan *recall*, yang menyeimbangkan kedua metrik tersebut:

$$F1\text{-score} = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (4)$$

Karena dataset bersifat multikelas dan tidak sepenuhnya seimbang antarkelas, keempat metrik di atas dihitung terlebih dahulu untuk masing-masing kelas, kemudian dirata-ratakan secara tertimbang (*weighted average*) berdasarkan jumlah sampel tiap kelas untuk memperoleh nilai *precision*, *recall*, dan F1-score keseluruhan, sebagaimana ditunjukkan pada Persamaan (5):

$$Metrik_{tertimbang} = \sum_{i=1}^n \left( \frac{n_i}{N} \times Metrik_i \right) \quad (5)$$

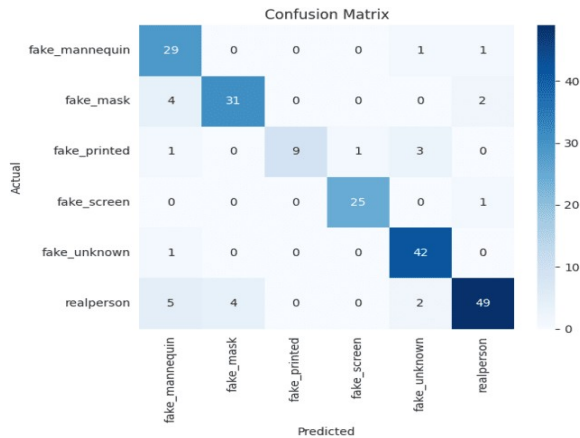
dengan  $n_i$  adalah jumlah sampel pada kelas ke- $i$ ,  $N$  adalah jumlah total sampel data uji, dan  $n$  adalah jumlah kelas (dalam penelitian ini  $n = 6$ ).

Hasil evaluasi menunjukkan akurasi sebesar 87,68%, *precision* tertimbang 88,79%, *recall* tertimbang 87,68%, dan F1-score tertimbang 87,62%. Rincian performa untuk masing-masing kelas disajikan pada Tabel 5.

Tabel 5. Laporan Klasifikasi per Kelas pada Data Uji

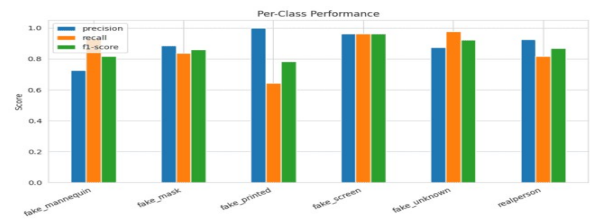
| Kelas          | Precision | Recall | F1-Score | N   |
|----------------|-----------|--------|----------|-----|
| fake_mannequin | 0,725     | 0,935  | 0,817    | 31  |
| fake_mask      | 0,886     | 0,838  | 0,861    | 37  |
| fake_printed   | 1,000     | 0,643  | 0,783    | 14  |
| fake_screen    | 0,962     | 0,962  | 0,962    | 26  |
| fake_unknown   | 0,875     | 0,977  | 0,923    | 43  |
| realperson     | 0,925     | 0,817  | 0,867    | 60  |
| Macro avg      | 0,895     | 0,862  | 0,869    | 211 |
| Weighted avg   | 0,888     | 0,877  | 0,876    | 211 |

Tabel 5 menunjukkan bahwa kelas *fake\_screen* dan *fake\_unknown* memperoleh performa terbaik dengan F1-score di atas 0,92, sedangkan kelas *fake\_printed* memiliki *precision* sempurna (1,000) namun *recall* terendah (0,643), yang berarti seluruh prediksi *fake\_printed* yang dikeluarkan model selalu benar, tetapi model masih melewatkan cukup banyak citra *fake\_printed* yang sebenarnya (false negative). Matriks konfusi pada Gambar 7 memberikan gambaran lebih rinci mengenai pola kesalahan klasifikasi antar kelas.



Gambar 7. Confusion matrix hasil klasifikasi model citra data uji.

Gambar 8 menyajikan perbandingan *precision*, *recall*, dan F1-score untuk setiap kelas secara berdampingan, yang memperjelas bahwa variasi performa antar kelas terutama dipengaruhi oleh *recall* pada kelas *fake\_printed* dan *fake\_mannequin*. Kelas *fake\_printed* memiliki *precision* yang sangat tinggi (mendekati 1,0) namun *recall* yang jauh lebih rendah (sekitar 0,65), yang mengindikasikan bahwa model cenderung sangat selektif dan jarang salah memprediksi citra lain sebagai *fake\_printed*, tetapi di sisi lain masih banyak citra *fake\_printed* asli yang justru terprediksi sebagai kelas lain. Pola serupa, meski tidak sekuat itu, juga terlihat pada kelas *fake\_mannequin*, di mana *precision*-nya relatif lebih rendah dibandingkan *recall*-nya, menunjukkan bahwa model cukup baik mengenali sebagian besar citra *fake\_mannequin*, tetapi juga cukup sering keliru mengklasifikasikan citra dari kelas lain sebagai *fake\_mannequin*. Sebaliknya, kelas *fake\_screen* dan *fake\_unknown* menunjukkan keseimbangan *precision* dan *recall* yang tinggi (di atas 0,90), yang mengindikasikan model mampu mengenali karakteristik kedua kelas ini secara konsisten. Kelas *fake\_mask* dan *realperson* berada pada posisi menengah, dengan *precision* dan *recall* yang relatif seimbang namun belum mencapai performa setinggi *fake\_screen* dan *fake\_unknown*. Secara keseluruhan, pola ini menunjukkan bahwa sumber utama kesalahan klasifikasi model lebih banyak berasal dari kebingungan dalam mengenali kelas *fake\_printed* dan *fake\_mannequin*, yang kemungkinan disebabkan oleh kemiripan visual antara kedua modalitas serangan tersebut dengan kelas lain.



Gambar 8. Perbandingan *precision*, *recall*, dan F1-score untuk setiap kelas.

### 3.3. Analisis Kesalahan Klasifikasi

Dari citra data uji, terdapat 26 citra (12,3%) yang diklasifikasikan secara keliru oleh model. Berdasarkan kelas sebenarnya, kesalahan paling banyak terjadi pada kelas *realperson* (11 kesalahan), diikuti *fake\_mask* (6 kesalahan) dan *fake\_printed* (5 kesalahan), sedangkan kelas *fake\_screen* dan *fake\_unknown* masing-masing hanya mengalami 1 kesalahan. Pasangan kelas yang paling sering tertukar adalah *realperson* yang diprediksi sebagai *fake\_mannequin* (5 kasus), *fake\_mask* yang diprediksi sebagai *fake\_mannequin* (4 kasus), dan *realperson* yang diprediksi sebagai *fake\_mask* (4 kasus). Contoh citra yang salah diklasifikasikan ditunjukkan pada Gambar 9.



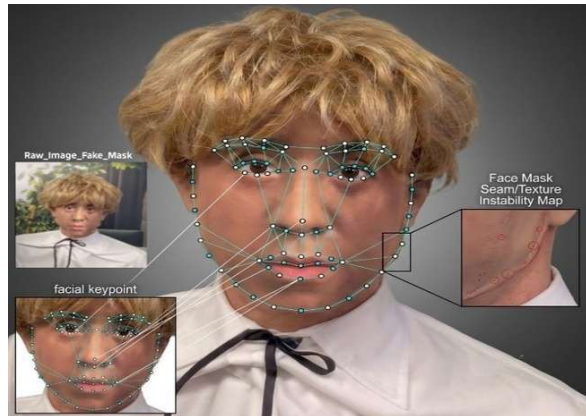
Gambar 9. Contoh citra yang salah diklasifikasikan

Pola kesalahan ini mengindikasikan bahwa model mengalami kesulitan utama dalam membedakan wajah asli (*realperson*) dari manekin (*fake\_mannequin*) maupun masker (*fake\_mask*), kemungkinan disebabkan oleh kemiripan tekstur kulit pada manekin dan masker berkualitas tinggi yang menyerupai kulit manusia pada pencahayaan tertentu [23], sebagaimana juga tercermin pada *precision* *fake\_mannequin* yang relatif rendah (0,725) akibat banyaknya citra dari kelas lain yang ikut diprediksi sebagai *fake\_mannequin*. Sebaliknya, kelas *fake\_printed* yang seharusnya mudah dikenali dari pola pantulan cahaya pada permukaan cetak [8], justru memiliki *recall* terendah, kemungkinan karena jumlah sampel pada kelas ini paling sedikit (98 citra) sehingga model kurang memiliki variasi contoh untuk mempelajari karakteristiknya secara menyeluruh [10].

Untuk mengilustrasikan lebih lanjut sumber kebingungan model pada kelas *fake\_mask*, dilakukan visualisasi titik-titik fitur wajah (*facial keypoints*) pada salah satu citra *fake\_mask*, sebagaimana ditunjukkan pada Gambar 10. Titik-titik *keypoint* pada area alis, mata, hidung, dan mulut (ditandai warna putih dan hijau kebiruan) menunjukkan bahwa struktur geometris wajah pada topeng tetap terdeteksi dengan baik oleh model, sehingga secara struktural sulit dibedakan dari wajah asli.

Namun demikian, perbesaran pada area batas topeng di sekitar rahang dan leher (*Face Mask Seam/Texture Instability Map*) memperlihatkan ketidakstabilan tekstur berupa sambungan (*seam*), perbedaan gradasi warna kulit, dan artefak permukaan yang tidak konsisten dengan tekstur kulit manusia asli. Area-area inilah yang menjadi kandidat sumber utama kesalahan klasifikasi model, karena pada level fitur global (bentuk dan tata letak wajah) topeng berkualitas tinggi sangat menyerupai wajah asli,

sedangkan perbedaan yang paling diskriminatif justru terletak pada detail tekstur lokal di area sambungan yang berukuran relatif kecil dan mudah terabaikan oleh model.



Gambar 10. Contoh visualisasi *facial keypoints* pada citra wajah yang menjadi indikasi sumber kesalahan klasifikasi

### 3.4. Ringkasan Karakteristik Model

Selain metrik akurasi, kelayakan model deteksi *presentation attack* untuk diterapkan pada sistem produksi juga ditentukan oleh karakteristik praktis seperti ukuran model, kecepatan inferensi, dan kebutuhan sumber daya komputasi, karena aspek-aspek ini menentukan apakah model dapat diintegrasikan secara *real-time* pada infrastruktur umum tanpa perangkat akselerasi khusus. Beberapa karakteristik praktis model tersebut turut dievaluasi untuk menilai kelayakan model diterapkan pada sistem

otentikasi biometrik wajah secara langsung, sebagaimana dirangkum pada Tabel 6.

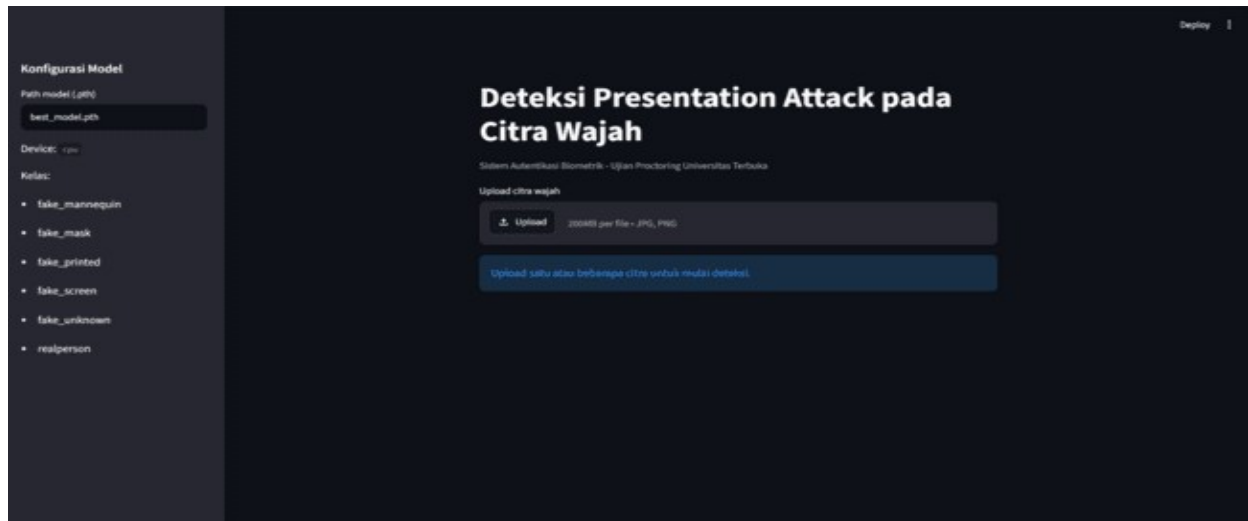
Tabel 6. Ringkasan Karakteristik Model Terbaik

| Karakteristik                 | Nilai               |
|-------------------------------|---------------------|
| Akurasi data uji              | 87,68%              |
| F1-score tertimbang           | 87,62%              |
| <i>Epoch</i> terbaik          | 24 dari 31          |
| Total waktu pelatihan         | 13,3 menit          |
| Waktu inferensi rata-rata     | 58,0 ms/citra (CPU) |
| Ukuran model (best_model.pth) | 27,13 MB            |
| Total parameter               | 6.960.006           |
| Parameter <i>trainable</i>    | 2.166.278 (31,12%)  |

Waktu inferensi rata-rata sebesar 58 milidetik per citra pada perangkat CPU serta ukuran model yang relatif ringkas (27,13 MB) menunjukkan bahwa model ini cukup ringan untuk diintegrasikan ke dalam sistem proctoring berbasis web tanpa membutuhkan akselerator GPU pada sisi server, sehingga lebih mudah diadopsi pada infrastruktur perbankan digital, layanan pemerintahan, hingga pendidikan jarak jauh pada umumnya.

### 3.5. Implementasi Antarmuka Deteksi Berbasis Streamlit

Model terbaik (best\_model.pth) yang diperoleh dari proses pelatihan selanjutnya diimplementasikan ke dalam sebuah aplikasi prototipe berbasis Streamlit untuk mendemonstrasikan penggunaan model secara interaktif sebagai modul deteksi *presentation attack*. Tampilan utama aplikasi ditunjukkan pada Gambar 11.



Gambar 11. Tampilan utama aplikasi deteksi *presentation attack* berbasis Streamlit.

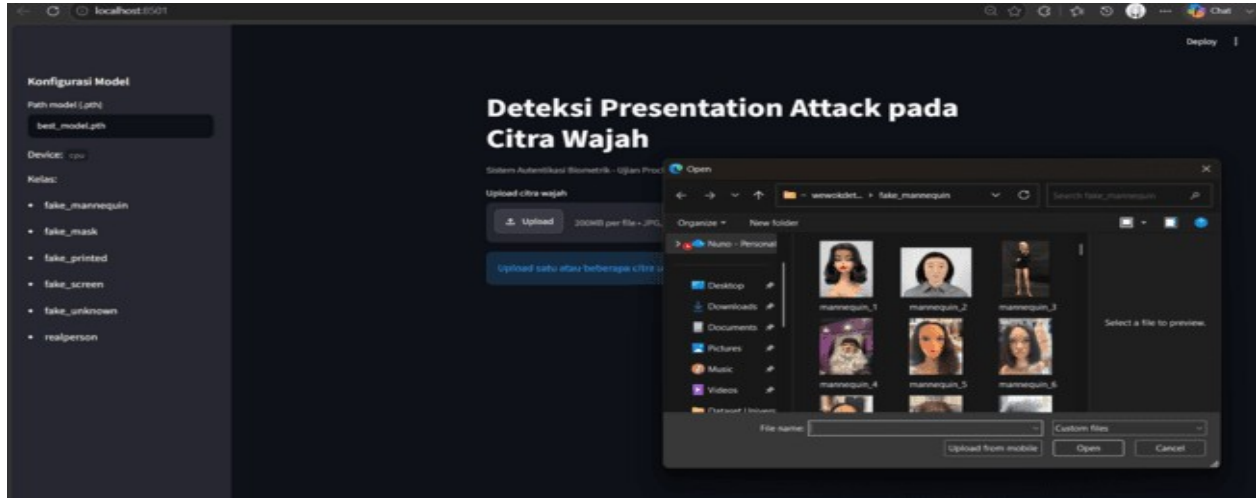
Sebagaimana terlihat pada Gambar 11, antarmuka aplikasi terbagi menjadi dua bagian utama yang dirancang untuk memudahkan penggunaan oleh operator atau pengawas ujian tanpa memerlukan keahlian teknis mendalam. Panel sebelah kiri (*sidebar*) berisi konfigurasi model, yaitu *path* berkas bobot model (best\_model.pth), indikator perangkat komputasi yang digunakan (device: cpu), serta daftar enam kelas keluaran model (fake\_mannequin, fake\_mask, fake\_printed, fake\_screen, fake\_unknown, dan realperson). Desain *sidebar* ini memungkinkan pengguna untuk memverifikasi secara langsung

bahwa model yang dimuat sudah benar sebelum proses deteksi dimulai, sekaligus memberikan transparansi mengenai kelas-kelas yang dikenali oleh sistem [22].

Panel utama menampilkan judul aplikasi "*Deteksi Presentation Attack pada Citra Wajah*" dengan subjudul "*Sistem Autentikasi Biometrik untuk Ujian Daring*", diikuti area unggah citra wajah yang mendukung format JPG dan PNG dengan ukuran maksimum 200 MB per berkas, serta pesan informatif yang mengarahkan pengguna untuk mengunggah satu atau beberapa

citra guna memulai proses deteksi. Pemilihan Streamlit sebagai *framework* antarmuka didasarkan pada kemampuannya menghasilkan aplikasi web interaktif berbasis Python secara cepat tanpa memerlukan pengembangan *front-end* terpisah, sehingga sangat sesuai untuk prototipe modul deteksi yang terintegrasi langsung dengan pipeline model PyTorch [22].

Proses pengunggahan citra dilakukan melalui dialog pemilihan berkas pada sistem operasi, seperti ditunjukkan pada Gambar 12, yang memperlihatkan bahwa aplikasi mendukung pengunggahan beberapa citra sekaligus (*multiple file upload*) dari folder dataset dalam contoh ini folder *fake\_mannequin* yang berisi sejumlah citra manekin (*mannequin\_1* hingga *mannequin\_6* dan seterusnya).



Gambar 12. Dialog pemilihan berkas pada antarmuka unggah citra wajah dengan dukungan unggah berkas ganda.

Hasil deteksi dari aplikasi setelah model dijalankan terhadap beberapa contoh citra ditunjukkan pada Gambar 13. Setiap citra yang diunggah ditampilkan berdampingan dengan kotak keterangan hasil prediksi beserta grafik batang probabilitas untuk keenam kelas, sehingga pengguna dapat melihat tidak hanya label akhir, tetapi juga tingkat keyakinan model terhadap setiap kemungkinan kelas.

Pada contoh pertama (*mannequin\_12.jpg*), aplikasi menampilkan kotak keterangan berwarna merah dengan teks "Presentation Attack Terdeteksi: fake\_mannequin" dan tingkat keyakinan

Kemampuan unggah ganda ini penting dalam konteks evaluasi sistem, karena memungkinkan pengawas atau peneliti untuk menguji model terhadap sekumpulan citra secara sekaligus dalam satu sesi tanpa harus mengunggah satu per satu [1]. Setiap berkas yang diunggah akan langsung diproses secara berurutan oleh model, sehingga hasil deteksi dapat diperoleh dengan cepat bahkan untuk sejumlah besar citra dalam sekali pengujian. Mekanisme ini juga mencerminkan skenario penggunaan nyata pada sistem proctoring, di mana beberapa *frame* tangkapan wajah dari satu sesi ujian mungkin perlu diverifikasi secara bersamaan untuk memastikan konsistensi identitas peserta sepanjang durasi ujian berlangsung [3].

95,3%, yang didukung oleh grafik batang probabilitas yang menunjukkan nilai mendekati 1,0 pada kelas *fake\_mannequin* dan mendekati 0 pada kelas lainnya. Pola serupa terlihat pada contoh kedua (*fakemask\_6.jpg*) yang dikenali sebagai *fake\_mask* dengan keyakinan 98,9%, dan contoh keempat (*unknown\_049.jpeg*) yang dikenali sebagai *fake\_unknown* dengan keyakinan 99,1%, di mana pada kedua kasus tersebut grafik probabilitas juga menunjukkan dominasi yang sangat tinggi pada satu kelas saja.



Gambar 13. Contoh hasil klasifikasi lima citra wajah pada aplikasi Streamlit, mencakup empat kasus presentation attack

Contoh ketiga (*fakeprinted\_19.jpg*) dikenali sebagai *fake\_printed* dengan keyakinan 89,9%; grafik probabilitasnya memperlihatkan

distribusi yang sedikit lebih tersebar, dengan kelas *fake\_printed* tetap dominan namun disertai probabilitas kecil pada kelas

fake\_mask. Pola ini sejalan dengan temuan pada Tabel 5 dan analisis kesalahan klasifikasi sebelumnya, yang menunjukkan bahwa kelas fake\_printed memiliki *recall* paling rendah di antara seluruh kelas, sehingga keyakinan model terhadap kelas ini relatif tidak setegas keyakinannya pada kelas fake\_mask maupun fake\_unknown, meskipun pada contoh kasus ini prediksinya tetap benar.

Contoh kelima (realperson\_31.jpg) merupakan satu-satunya kasus pada Gambar 13 yang menampilkan kotak keterangan berwarna hijau dengan teks "Wajah Asli (Real Person)" dan tingkat keyakinan 98,0%, yang membedakannya secara visual dari keempat contoh lain yang seluruhnya merupakan kasus *presentation attack* dengan kotak berwarna merah. Perbedaan warna kotak keterangan ini (merah untuk *presentation attack*, hijau untuk wajah asli) merupakan elemen penting pada antarmuka karena memungkinkan pengguna sistem autentikasi mengambil keputusan secara cepat hanya dengan melihat indikator warna, tanpa perlu menginterpretasikan nilai probabilitas secara manual.

Secara keseluruhan, kelima contoh pada Gambar 13 menggambarkan bahwa model mampu memberikan prediksi dengan tingkat keyakinan tinggi (di atas 89%) pada berbagai jenis *presentation attack* yang berbeda karakteristik visualnya, mulai dari manekin tiga dimensi, masker wajah, hingga citra hasil pemotretan ulang permukaan cetak, sekaligus tetap mampu mengenali wajah asli dengan keyakinan yang sebanding. Hasil demonstrasi ini konsisten dengan metrik evaluasi kuantitatif pada Tabel 5 dan memperkuat kelayakan model untuk diujicobakan lebih lanjut sebagai komponen autentikasi biometrik pada sistem verifikasi wajah berskala besar.

#### 4. KESIMPULAN

Penelitian ini menghasilkan model deteksi *presentation attack* berbasis transfer learning dengan fine-tuning parsial pada arsitektur DenseNet121, dilatih pada dataset enam kelas berjumlah 1.403 citra. Model mencapai akurasi 87,68%, precision tertimbang 88,79%, recall tertimbang 87,68%, dan F1-score tertimbang 87,62%, dengan waktu inferensi rata-rata 58 ms per citra sehingga layak dioperasikan pada perangkat keras standar. Kesalahan klasifikasi paling sering terjadi antara kelas realperson, fake\_mannequin, dan fake\_mask, sementara fake\_printed memiliki recall terendah akibat jumlah sampel paling sedikit.

Model terbaik diimplementasikan ke prototipe aplikasi Streamlit yang mampu mendeteksi secara interaktif, memperkuat kompatibilitasnya dengan infrastruktur berdaya rendah. Pendekatan fine-tuning parsial terbukti efektif mengadaptasi fitur pra-latih ImageNet untuk tugas klasifikasi domain khusus ini, menjadi alternatif yang lebih efisien dibanding fine-tuning penuh maupun penambahan modul seperti channel attention, tanpa mengorbankan akurasi secara signifikan. Secara keseluruhan, DenseNet121 dengan fine-tuning parsial menghadirkan keseimbangan antara akurasi, efisiensi, dan kemudahan implementasi, menjadikannya kandidat layak sebagai lapisan deteksi *presentation attack* pada sistem proctoring ujian daring di lingkungan Pendidikan Jarak Jauh.

Penelitian ini memiliki beberapa limitasi. Pertama, ketidakseimbangan kelas pada dataset (rasio 4,07 antara realperson dan fake\_printed) belum ditangani secara eksplisit; *CrossEntropyLoss* digunakan tanpa pembobotan kelas, oversampling, atau teknik penanganan imbalance lain, hanya diimbangi secara tidak langsung melalui stratified split dan metrik tertimbang, yang kemungkinan berkontribusi pada rendahnya recall kelas fake\_printed. Kedua, ukuran dataset yang terbatas (1.403 citra) menjadi salah satu faktor di balik kecenderungan overfitting ringan yang teramati selama pelatihan. Ketiga, dataset merupakan kompilasi dari berbagai sumber publik dengan kondisi pengambilan citra yang tidak sepenuhnya terkontrol, sehingga generalisasi model pada kondisi dunia nyata seperti variasi pencahayaan, sudut pengambilan, dan jenis kamera belum diuji secara khusus. Keempat, evaluasi model juga masih terbatas pada modalitas serangan yang tercakup dalam enam kelas dataset, sehingga ketahanan model terhadap jenis *presentation attack* baru yang belum pernah dijumpai sebelumnya (unseen attack) belum dapat dipastikan. Berdasarkan limitasi tersebut, riset selanjutnya dapat mengeksplorasi penanganan ketidakseimbangan kelas secara eksplisit (class weighting, focal loss, atau oversampling) untuk meningkatkan recall kelas minoritas, penambahan volume dan variasi data disertai cross-dataset evaluation untuk menguji generalisasi, serta integrasi mekanisme attention pada area tekstur lokal yang menjadi sumber kesalahan. Selain itu, pengujian pada perangkat mobile/edge, evaluasi ketahanan model terhadap serangan adversarial, serta eksplorasi kemampuan model dalam mendeteksi modalitas serangan baru di luar enam kelas yang tersedia juga menjadi arah pengembangan yang relevan untuk memperkuat kesiapan model diterapkan pada sistem proctoring ujian daring secara nyata.

#### DAFTAR PUSTAKA

- [1] A. Putri Iskandar, Muhammad Ikhsan Thohir, Ivana Lucia Kharisma, Kamdan, and Anggun Fergina, "Implementasi Deteksi Langsung Pada Sistem Ujian Online Menggunakan Algoritma Convolutional Neural Network," *Jurnal CoSciTech (Computer Science and Information Technology)*, vol. 5, no. 2, pp. 483–492, Sep. 2024, doi: <https://doi.org/10.37859/coscitech.v5i2.7270>.
- [2] Muhammad Arsal, Bheta Agus Wardijono, and D. Anggraini, "Face Recognition Untuk Akses Pegawai Bank Menggunakan Deep Learning Dengan Metode CNN," *Jurnal Nasional Teknologi dan Sistem Informasi*, vol. 6, no. 1, pp. 55–63, Jun. 2020, doi: [10.25077/teknosi.v6i1.2020.55-63](https://doi.org/10.25077/teknosi.v6i1.2020.55-63).
- [3] E. N. F. Dewi, G. N. Tarempa, A. N. Rachman, and A. Febriansyah, "Sistem Presensi Perkuliahan Mahasiswa Berbasis Face Recognition dengan Metode Liveness Detection," *Indonesian Journal Computer Science*, vol. 4, no. 2, pp. 118–126, Oct. 2025, doi: [10.31294/ijcs.v4i2.9876](https://doi.org/10.31294/ijcs.v4i2.9876).
- [4] N. Selwyn, C. O'Neill, G. Smith, M. Andrejevic, and X. Gu, "A necessary evil? The rise of online exam proctoring in Australian universities," *Media Int. Aust.*, vol. 186, no. 1, pp. 149–164, 2023, doi: [10.1177/1329878X211005862](https://doi.org/10.1177/1329878X211005862).

- [5] M. Pooshideh, "A novel active solution for two-dimensional face presentation attack detection," *arXiv preprint*, arXiv:2212.06958, 2022. Available: <https://arxiv.org/abs/2212.06958>
- [6] W. Yaqub, M. Mohanty, and B. Suleiman, "Image-hashing-based anomaly detection for privacy-preserving online proctoring," *arXiv preprint*, arXiv:2107.09373, 2021. Available: <https://arxiv.org/abs/2107.09373>
- [7] Z. Boulkenafet, J. Komulainen, and A. Hadid, "Face Anti-Spoofing Based on Color Texture Analysis," *arXiv preprint*, arXiv:1511.06316, 2015. Available: <https://arxiv.org/abs/1511.06316>
- [8] X. Tan, Y. Li, J. Liu, and L. Jiang, "Face liveness detection from a single image with sparse low rank bilinear discriminative model," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Crete, Greece, 2010, pp. 504–517, doi: [10.1007/978-3-642-15567-3\\_37](https://doi.org/10.1007/978-3-642-15567-3_37)
- [9] G. Pan, L. Sun, Z. Wu, and S. Lao, "Eyeblick-based anti-spoofing in face recognition from a generic webcam," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Rio de Janeiro, Brazil, 2007, pp. 1–8, doi: [10.1109/ICCV.2007.4409068](https://doi.org/10.1109/ICCV.2007.4409068)
- [10] O. Nikisins, A. Mohammadi, A. Anjos, and S. Marcel, "On effectiveness of anomaly detection approaches against unseen presentation attacks in face anti-spoofing," in *Proc. Int. Conf. Biometrics (ICB)*, Gold Coast, Australia, 2018, pp. 75–81, doi: [10.1109/ICB2018.2018.00022](https://doi.org/10.1109/ICB2018.2018.00022)
- [11] Y. Liu, A. Jourabloo, and X. Liu, "Learning deep models for face anti-spoofing: Binary or auxiliary supervision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Salt Lake City, UT, USA, 2018, pp. 389–398, doi: [10.1109/CVPR.2018.00048](https://doi.org/10.1109/CVPR.2018.00048)
- [12] R. A. Saragih, T. D. Sunoto, D. Setiadarunia, and E. Moses, "Model Deep Learning untuk Face Anti-Spoofing dalam Mengatasi Domain Generalization dengan Depth Estimation dan Generative Adversarial Network," *Jurnal Telematika*, vol. 20, no. 1, pp. 1–12, Aug. 2025, doi: [10.61769/telematika.v20i1.730](https://doi.org/10.61769/telematika.v20i1.730). Available: <https://journal.ithb.ac.id/index.php/telematika/article/view/730>
- [13] Dzikri Maulana, Nur Salamah Azzahrah, Reiza Alithian Syach, Sopian Syaury, and Chaerur Rozikin, "Identifikasi Konten Visual Buatan Ai Dengan RESNET Dan Fine-Grained Feature Extraction," *Jurnal Informatika dan Teknik Elektro Terapan*, vol. 13, no. 3S1, Oct. 2025, doi: [10.23960/jitet.v13i3S1.8056](https://doi.org/10.23960/jitet.v13i3S1.8056)
- [14] I. Made and N. Devi, "Real Time Face Recognition for Mobile Application Based on Mobilenetv2," *Jurnal Multidisiplin Madani*, vol. 3, no. 9, pp. 1855–1864, Sep. 2023, doi: [10.55927/mudima.v3i9.5924](https://doi.org/10.55927/mudima.v3i9.5924)
- [15] F. A. Jiwani and Bagus Satrio Waluyo Poetro, "Sistem Deteksi Gambar Deepfake Menggunakan Cnn Densenet-121 Dengan Watermarking Least Significant Bit (LSB)," *Jurnal Rekayasa Sistem Informasi dan Teknologi*, vol. 2, no. 3, pp. 1157–1170, Feb. 2025, doi: [10.70248/jrsit.v2i3.1939](https://doi.org/10.70248/jrsit.v2i3.1939)
- [16] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely Connected Convolutional Networks," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2261–2269, Jul. 2017, doi: [10.1109/CVPR.2017.243](https://doi.org/10.1109/CVPR.2017.243)
- [17] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Miami, FL, USA, 2009, pp. 248–255, doi: [10.1109/CVPR.2009.5206848](https://doi.org/10.1109/CVPR.2009.5206848)
- [18] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014. Available: <https://jmlr.org/papers/v15/srivastava14a.html>
- [19] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, 2016, pp. 770–778, doi: [10.1109/CVPR.2016.90](https://doi.org/10.1109/CVPR.2016.90)
- [20] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, New Orleans, LA, USA, 2019. Available: <https://arxiv.org/abs/1711.05101>
- [21] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, San Diego, CA, USA, 2015. Available: <https://arxiv.org/abs/1412.6980>
- [22] A. Paszke *et al.*, "PyTorch: An imperative style, high-performance deep learning library," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, Vancouver, BC, Canada, 2019, pp. 8024–8035. Available: <https://arxiv.org/abs/1912.01703>
- [23] A. Anwar and A. Raychowdhury, "Masked face recognition for secured authentication," *arXiv preprint*, arXiv:2008.11104, 2020. Available: <https://arxiv.org/abs/2008.11104>
- [24] A. Dosovitskiy *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, Virtual, 2021. Available: <https://arxiv.org/abs/2010.11929>
- [25] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. Int. Conf. Mach. Learn. (ICML)*, Long Beach, CA, USA, 2019, pp. 6105–6114. Available: <https://arxiv.org/abs/1905.11946>
- [26] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, 2016, pp. 2818–2826, doi: [10.1109/CVPR.2016.308](https://doi.org/10.1109/CVPR.2016.308)
- [27] S. D. Thepade, M. Dindorkar, P. Chaudhari, and S. Bang, "Face presentation attack identification optimization with adjusting convolution blocks in VGG networks," *Intell. Syst. Appl.*, vol. 16, 2022, Art. no. 200107, doi: [10.1016/j.iswa.2022.200107](https://doi.org/10.1016/j.iswa.2022.200107)