Journal of System & Technology
ISSN: 3110-0171
Vol. 2, No. 1, June 2026

PERBANDINGAN KINERJA MODEL CNN EFFICIENTNETB0 DAN VISION TRANSFORMER UNTUK KLASIFIKASI CITRA REAL-FAKE

Akhlaqul Muhammad Fadwa[1], M. Sohibbal[1], Rachel Ditya Syaharani[1], Dian Ramadhani[1]*, Al Aminuddin[2]
[1]Dept.of Informatics Engineering, Faculty of Engineering, Universitas Riau, Pekanbaru, Indonesia
[2]Sistem Informasi, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Riau, Pekanbaru, Indonesia
akhlaqul.muhammad2834@student.unri.ac.id, m.sohibbal5312@student.unri.ac.id,
rachel.ditya6367@student.unri.ac.id, dianramadhani@lecturer.unri.ac.id, al.aminuddin@lecturer.unri.ac.id
*Corresponding author: dianramadhani@lecturer.unri.ac.id

Abstract: Model generatif berbasis Generative Adversarial Network (GAN) seperti StyleGAN, Stable Diffusion, DALL·E, dan DeepFaceLab kini mampu menghasilkan citra sintetis yang sulit dibedakan dari citra asli, memicu penyalahgunaan berupa deepfake, manipulasi identitas digital, dan misinformasi visual yang mengancam kepercayaan terhadap konten digital. Penelitian ini mengusulkan pendekatan komparatif antara Convolutional Neural Network (CNN) berbasis EfficientNetB0 dan Vision Transformer (ViT) untuk mengklasifikasikan citra real dan fake guna mengidentifikasi arsitektur paling efektif dalam deteksi konten sintetis. Kedua model dikonfigurasi dengan backbone frozen menggunakan bobot pre-trained dan classifier head identik: Dense(512, ReLU), Dropout(0,4), Dense(256, ReLU), Dropout(0,4), Dense(1, Sigmoid), untuk memastikan perbandingan yang objektif. Dataset CIFAKE terdiri dari 120.000 citra — 60.000 real (CIFAR-10) dan 60.000 fake (Stable Diffusion v1.4) — diproses dengan resizing ke 224×224 piksel, normalisasi, dan augmentasi data, dengan pembagian 80:20 untuk pelatihan-validasi dan 20.000 citra untuk pengujian. Evaluasi dilakukan menggunakan akurasi, precision, recall, F1-score, dan confusion matrix. EfficientNetB0 mencapai akurasi 84,79% dengan waktu pelatihan ±1 jam 12 menit, namun lemah pada deteksi kelas fake dengan recall 72,17% dan false positive 2.783 citra. Sebaliknya, Vision Transformer mencapai akurasi 97,15% dengan precision, recall, dan F1-score seimbang di atas 97% pada kedua kelas, dengan waktu pelatihan ±14 jam 18 menit. Vision Transformer terbukti mengungguli EfficientNetB0 pada seluruh metrik, sementara EfficientNetB0 tetap relevan pada skenario dengan keterbatasan komputasi. Secara khusus, untuk menekan angka false positive pada EfficientNetB0, penelitian selanjutnya dapat menerapkan class-weighted loss atau focal loss guna meningkatkan sensitivitas model terhadap kelas fake, serta threshold tuning pada output sigmoid untuk menyeimbangkan precision dan recall.

Keywords—CIFAKE, Deep Learning, EfficientNetB0, Klasifikasi Citra, Vision Transformer

Article Information:
Received: 09-Jun-2026; Revised: 21-Jun-2026; Accepted: 26-Jun-2026; Published: 30-Jun-2026

This work is licensed under a **CC BY-SA**. Copyright ©2026 by Author. Published by Universitas Riau.

I. PENDAHULUAN

Perkembangan teknologi kecerdasan buatan pada bidang *computer vision* telah membawa perubahan signifikan dalam beberapa tahun terakhir, terutama melalui kemampuan model generatif untuk menghasilkan citra sintetis yang sangat realistis. Teknik seperti Generative Adversarial Network (GAN) dan *latent diffusion model* menjadi tulang punggung dari kemampuan tersebut, menghasilkan citra yang sering kali tidak dapat dibedakan dari citra asli secara visual oleh manusia [1]. Konsekuensi langsung dari kemajuan ini adalah munculnya berbagai bentuk penyalahgunaan, mulai dari penyebaran informasi palsu, manipulasi konten visual untuk kepentingan politik, hingga penipuan berbasis identitas digital [2].

Skala permasalahan ini terus meningkat secara signifikan. Laporan industri mencatat lonjakan jumlah file *deepfake* yang drastis dalam dua tahun terakhir, disertai dengan peningkatan kerugian finansial akibat insiden *deepfake* serta penurunan kemampuan manusia dalam mendeteksi konten manipulatif berkualitas tinggi [3]. Kondisi ini menegaskan kebutuhan mendesak akan sistem deteksi otomatis yang akurat, efisien, dan mampu beradaptasi dengan perkembangan model generatif yang semakin kompleks.

Salah satu pendekatan yang paling banyak dimanfaatkan untuk persoalan ini adalah *deep learning*, khususnya Convolutional Neural Network (CNN) yang telah terbukti efektif dalam berbagai tugas klasifikasi citra [4][5]. Seiring waktu, muncul varian CNN yang lebih efisien seperti EfficientNet yang memperkenalkan konsep *compound scaling*, yaitu metode penyeimbangan kedalaman, lebar, dan resolusi input secara simultan untuk memperoleh keseimbangan optimal antara akurasi dan efisiensi komputasi [6]. EfficientNetB0 sebagai model dasar dari keluarga EfficientNet menawarkan performa kompetitif dengan jumlah parameter yang relatif kecil (~5,3 juta parameter), sehingga banyak digunakan sebagai baseline pada berbagai tugas klasifikasi citra modern [7]. Karakteristik ini menjadikan EfficientNetB0 representasi ideal dari arsitektur berbasis fitur lokal (CNN) yang mengutamakan efisiensi komputasi.

Di sisi lain, pendekatan berbasis Transformer yang

awalnya dikembangkan untuk pemrosesan bahasa alami juga telah berhasil diadaptasi ke bidang visi komputer melalui model Vision Transformer (ViT) yang diperkenalkan oleh Dosovitskiy et al. [8]. Berbeda dengan CNN yang melakukan ekstraksi fitur secara lokal, ViT membagi citra menjadi sejumlah *patch* yang kemudian diproses sebagai urutan data menggunakan mekanisme *self-attention*. Pendekatan ini memungkinkan model menangkap hubungan antar bagian citra secara global, dan dalam beberapa skenario menghasilkan performa yang lebih baik dibandingkan CNN konvensional [9]. Meskipun demikian, ViT-Base memerlukan sumber daya komputasi yang lebih besar dibandingkan CNN, sehingga menjadikannya representasi arsitektur berbasis fitur global (Transformer) yang menekankan kualitas ekstraksi fitur di atas efisiensi.

Dataset CIFAKE menjadi salah satu *benchmark* yang relevan untuk menguji kemampuan kedua pendekatan tersebut. Dataset ini berisi 60.000 citra *real* yang berasal dari CIFAR-10 dan 60.000 citra sintetis yang dibangkitkan menggunakan Stable Diffusion v1.4, sehingga menyediakan distribusi yang seimbang antara dua kelas pada resolusi rendah 32×32 piksel [10]. Resolusi rendah ini menjadi tantangan tersendiri karena artefak visual yang biasa dimanfaatkan model deteksi menjadi lebih sulit dikenali, sehingga menuntut kemampuan model yang lebih tinggi dalam mengekstraksi fitur diskriminatif.

Berdasarkan latar belakang tersebut, rumusan masalah dalam penelitian ini meliputi: bagaimana implementasi EfficientNetB0 untuk klasifikasi citra *real* dan *fake* pada dataset CIFAKE, bagaimana implementasi Vision Transformer pada dataset yang sama, serta bagaimana perbandingan performa kedua model berdasarkan metrik evaluasi yang digunakan. Oleh karena itu, penelitian ini bertujuan untuk membandingkan kinerja EfficientNetB0 dan Vision Transformer dalam klasifikasi citra *real* dan *fake* pada dataset CIFAKE guna mengidentifikasi arsitektur yang paling efektif dalam mendeteksi konten sintetis.

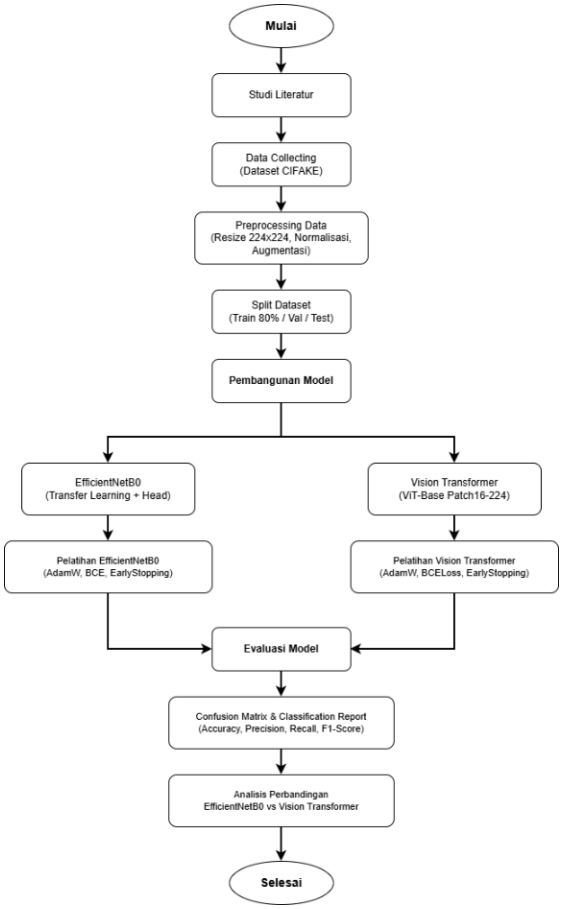
II. METODE PENELITIAN

Penelitian ini menggunakan pendekatan kuantitatif eksperimental dengan melatih dua model *deep learning* menggunakan dataset yang sama, untuk kemudian dibandingkan performanya secara objektif. Tahapan penelitian terdiri dari studi literatur, pengumpulan data, *preprocessing*, pembangunan model, pelatihan, hingga evaluasi sebagaimana ditunjukkan pada Gambar 1.

A. Studi Literatur

Studi literatur dilakukan untuk membangun landasan teori yang kuat dalam mendukung penelitian ini. Kajian pustaka mencakup perkembangan model generatif seperti GAN dan *latent diffusion* dalam pembangkitan citra sintetis, serta dampaknya terhadap penyebaran *deepfake* dan misinformasi visual [1][2]. Dikaji pula berbagai pendekatan deteksi citra manipulatif berbasis *deep learning*, mulai dari arsitektur CNN konvensional hingga varian yang lebih modern seperti EfficientNet dengan pendekatan *compound scaling* [6][7].

Sebagai pembanding terhadap pendekatan CNN, tinjauan terhadap Vision Transformer (ViT) juga dilakukan untuk memahami mekanisme self-attention dan keunggulannya dalam menangkap fitur global citra yang tidak dapat ditangkap secara lokal oleh konvolusi [8][9]. Untuk mendukung eksperimen perbandingan kedua arsitektur tersebut, referensi terkait dataset CIFAKE sebagai benchmark turut dikaji guna memahami karakteristik distribusi data, resolusi citra, serta tantangan klasifikasi yang ditawarkan [10]. Hasil studi literatur ini menjadi dasar pemilihan arsitektur, metode *preprocessing*, serta metrik evaluasi yang digunakan.



Gambar 1. Tahapan Penelitian

B. Data Collecting

Dataset yang digunakan dalam penelitian ini adalah CIFAKE yang tersedia secara publik di platform Kaggle [10]. Dataset ini terdiri dari citra berukuran 32×32 piksel dengan dua kelas utama, yaitu citra *real* yang bersumber dari CIFAR-10 dan citra *fake* yang dibangkitkan menggunakan Stable Diffusion v1.4. Total keseluruhan dataset berjumlah 120.000 citra dengan distribusi seimbang antara kedua kelas, masing-masing sebanyak 60.000 citra. Informasi lengkap dataset disajikan pada Tabel 1.

Tabel 1. Informasi Dataset CIFAKE

Atribut	Keterangan
Sumber Dataset	Kaggle (CIFAKE)
Jumlah Total Data	120.000 citra
Kelas	Real (CIFAR-10) dan Fake (Stable

Atribut	Keterangan
	Diffusion v1.4)
Resolusi Asli	32×32 piksel
Format	RGB (3 channel)
Distribusi Kelas	Seimbang (60.000 per kelas)

C. Preprocessing Data

Tahap preprocessing dilakukan untuk menyesuaikan data mentah agar sesuai dengan kebutuhan input model. Tanpa preprocessing, citra yang memiliki ukuran dan distribusi nilai piksel berbeda dapat memengaruhi kinerja model serta menurunkan akurasi klasifikasi. Proses ini juga membantu model mengenali pola dengan lebih baik, meningkatkan kemampuan generalisasi, serta mengurangi risiko overfitting pada data pelatihan.

Pada penelitian ini, seluruh citra di-resize dari ukuran asli 32×32 piksel menjadi 224×224 piksel menggunakan interpolasi bilinear karena ukuran tersebut merupakan input standar EfficientNetB0 dan ViT-Base Patch16-224. Selanjutnya dilakukan normalisasi untuk menyamakan distribusi nilai piksel sehingga proses pelatihan model menjadi lebih stabil.

Pada EfficientNetB0 digunakan fungsi *preprocess_input* dari Keras, sedangkan pada Vision Transformer digunakan normalisasi dengan mean dan standar deviasi sebesar 0,5 pada setiap channel RGB sehingga nilai piksel berada pada rentang [-1,1]. Proses normalisasi tersebut dirumuskan sebagai berikut: Transformer, normalisasi menggunakan *transforms.Compose()* dari torchvision dengan *mean* dan standar deviasi 0,5 pada setiap *channel* RGB, sehingga nilai piksel berada pada rentang [-1, 1] (persamaan 1).

$$x_{norm} = \frac{x - 0{,}5}{0{,}5}, x_{norm} \in [-1{,}1] \quad (1)$$

Selain itu, augmentasi data diterapkan pada data pelatihan menggunakan ImageDataGenerator dengan parameter width_shift_range, height_shift_range, dan zoom_range sebesar 0,2 untuk meningkatkan variasi data dan membantu model melakukan generalisasi dengan lebih baik [11]. Dataset kemudian dibagi menjadi data pelatihan, validasi, dan pengujian menggunakan *flow_from_directory*, *ImageFolder*, serta *random_split* dengan rasio 80:20 untuk train-validation.

D. Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) adalah arsitektur *deep learning* yang dirancang khusus untuk memproses data berstruktur grid seperti citra. CNN menggunakan operasi konvolusi untuk mengekstraksi fitur lokal secara otomatis. Secara matematis, operasi konvolusi pada *layer* ke-*l* dirumuskan sebagai [12] (persamaan 2).

$$y_{i,j}^{l} = f\left(\sum_{m}\sum_{n} w_{m,n}^{l} \cdot x_{i+m,\,j+n}^{l-1} + b^{l}\right) \quad (2)$$

di mana $w_{m,n}^{l}$adalah bobot filter, x^{l-1}adalah peta fitur dari layer sebelumnya, b^{l}adalah bias, dan $f(\cdot)$adalah fungsi aktivasi. Arsitektur CNN terdiri dari convolutional layer, pooling layer, dan fully connected layer yang bekerja secara bertahap dalam memproses citra[12].

E. Algoritma Agglomerative Hierarchical Clustering

Model ini terdiri dari sekitar 237 *layer* dengan sekitar 5,3 juta parameter dan menggunakan blok utama MBConv (*Mobile Inverted Bottleneck Convolution*) yang dikombinasikan dengan *squeeze-and-excitation* [6]. Dalam penelitian ini, EfficientNetB0 digunakan dengan pendekatan *transfer learning* dari ImageNet dengan *backbone* di-*freeze*. *Classifier head* yang ditambahkan disajikan pada Tabel 2.

Tabel 2. Arsitektur Classifier Head EfficientNetB0

Layer	Output Shape	Parameter
EfficientNetB0 (frozen backbone)	7×7×1280	~4,0 juta (frozen)
Global Average Pooling 2D	1280	0
Dense (ReLU, 512)	512	655,872
Dropout (rate=0,4)	512	0
Dense (ReLU, 256)	256	131,328
Dropout (rate=0,4)	256	0
Dense (Sigmoid, 1)	1	257
Total Trainable Parameters		787,457

F. Vision Transformer (ViT)

Vision Transformer (ViT) mengadaptasi arsitektur Transformer untuk visi komputer dengan membagi citra menjadi *patch* berukuran tetap P×P piksel [8]. Sebuah citra berukuran H×W dengan C *channel* warna dibagi menjadi sejumlah *patch* dua dimensi yang tidak tumpang tindih (*non-overlapping*). Jumlah *patch* N yang dihasilkan diperoleh dari persamaan 3:

$$N = \frac{H \times W}{P^2} \quad (3)$$

di mana N adalah jumlah *patch* yang dihasilkan, H adalah tinggi citra (*height*) dalam piksel, W adalah lebar citra (*width*) dalam piksel, dan P adalah ukuran sisi setiap *patch* dalam piksel. Pada penelitian ini digunakan ViT-Base Patch16-224, sehingga nilai P = 16 dan ukuran input H × W = 224 × 224, menghasilkan N = (224 × 224) / 16² = 196 patch per citra.

Setiap *patch* diproyeksikan ke ruang *embedding* berdimensi D (persamaan 4):

$$z_0 = [x_{class};\ x_p^1 E;\ x_p^2 E;\ \ldots;\ x_p^N E] + E_{pos} \quad (4)$$

di mana $E \in \mathbb{R}^{(P^2 \cdot C) \times D}$ adalah matriks proyeksi, x_{class}adalah token klasifikasi, dan E_{pos}adalah positional embedding [8].

Inti ViT adalah mekanisme *Multi-Head Self-Attention* (MHSA) yang memungkinkan setiap *patch* memperhatikan seluruh *patch* lain secara global (persamaan 5).

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (5)$$

di mana Q, K, V adalah proyeksi linear dari embedding input dan $d_k$adalah dimensi key [8].

Dalam penelitian ini digunakan ViT-Base Patch16-224 dari library Hugging Face Transformers [13], diimplementasikan dengan PyTorch. *Backbone* di-*freeze* seluruhnya agar perbandingan dilakukan pada kondisi setara. *Classifier head* yang ditambahkan disajikan pada Tabel 3.

Tabel 3. Arsitektur Classifier Head Vision Transformer

Layer	Output Shape	Parameter
ViT-Base Patch16-224 (frozen)	768	~86 juta (frozen)
Linear (ReLU, 512)	512	393,728
Dropout (rate=0,4)	512	0
Linear (ReLU, 256)	256	131,328
Dropout (rate=0,4)	256	0
Linear (Sigmoid, 1)	1	257
Total Trainable Parameters		525,313

G. *Confusion Matrix*

Model *Confusion matrix* adalah tabel evaluasi yang membandingkan hasil prediksi model terhadap label aktual, terdiri dari True Positive (TP), True Negative (TN), False Positive (FP), dan False Negative (FN) [14]. Dari komponen ini diturunkan empat metrik evaluasi akurasi (persamaan 6), precision (persamaan 7), recall (persamaan 8), dan F1-Score (persamaan 9).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (6)$$

$$Precision = \frac{TP}{TP + FP} \quad (7)$$

$$Recall = \frac{TP}{TP + FN} \quad (8)$$

$$F1\text{-}score = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (9)$$

Dalam konteks deteksi *deepfake*, *recall* pada kelas *fake* menjadi metrik yang sangat kritis karena mengukur kemampuan model mendeteksi seluruh citra *fake* yang sebenarnya. *False positive* yang tinggi — yakni citra *fake* yang salah diprediksi sebagai *real* — merupakan jenis kesalahan paling berisiko dalam aplikasi nyata [15].

III. Hasil dan Pembahasan

Penelitian ini dimulai dengan pengumpulan dataset CIFAKE dari Kaggle yang terdiri dari citra real dan fake. Selanjutnya dilakukan tahap preprocessing untuk memastikan data bersih, terstruktur, dan siap digunakan dalam proses pelatihan model. Tahap ini bertujuan meningkatkan kualitas data sehingga model dapat mengenali pola dengan lebih baik serta memiliki kemampuan generalisasi yang optimal.

Setelah preprocessing, dilakukan proses klasifikasi menggunakan dua model deep learning untuk mendeteksi citra real dan fake. Kinerja kedua model kemudian dibandingkan berdasarkan hasil evaluasi untuk mengetahui model yang memiliki performa terbaik dalam klasifikasi citra.

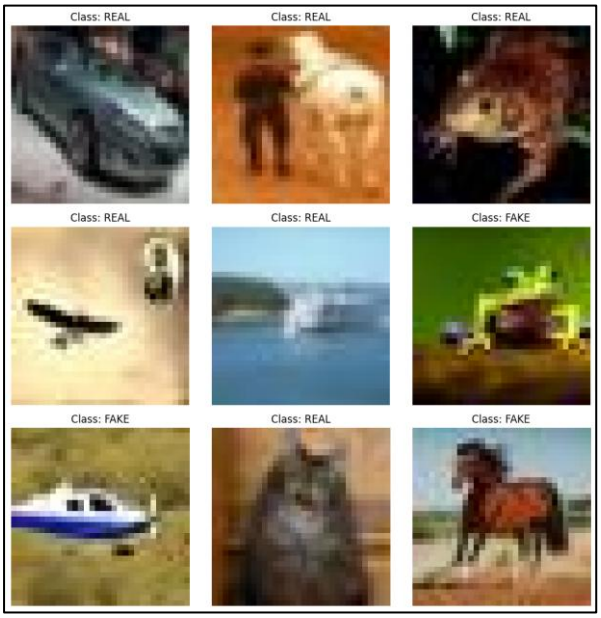
A. *Data Collecting*

Dataset Dataset CIFAKE diperoleh dari Kaggle dan terdiri dari 120.000 citra 32×32 piksel yang terbagi merata ke dalam dua kelas: *real* (CIFAR-10) dan *fake* (Stable Diffusion v1.4). Distribusi kelas yang seimbang menjadikan dataset ini ideal untuk pelatihan model klasifikasi biner tanpa bias kelas. Distribusi data disajikan pada Tabel 4.

Tabel 4. Distribusi Kelas Dataset CIFAKE

Kelas	Jumlah Citra	Persentase
Real (CIFAR-10)	60.000	50%
Fake (Stable Diffusion v1.4)	60.000	50%
Total	120.000	50%

Berikut cuplikan 9 data citra pada dataset CIFAKE disajikan pada Gambar 2.



Gambar 2. Cuplikan 9 sampel data dari dataset CIFAKE

B. *Data Preprocessing*

Proses *preprocessing* dilakukan untuk mempersiapkan data sebelum digunakan dalam pelatihan. Seluruh citra di-*resize* dari 32×32 menjadi 224×224 piksel. Untuk EfficientNetB0, normalisasi menggunakan preprocess_input melalui ImageDataGenerator dengan augmentasi *width/height shift* 0,2 dan *zoom* 0,2. Untuk ViT, normalisasi menggunakan transforms.Compose() dengan *mean* dan *std* 0,5 per *channel* RGB sehingga nilai piksel berada di rentang [-1, 1], serta *splitting* menggunakan ImageFolder dan random_split rasio 80:20. Pembagian dataset untuk pelatihan dan pengujian disajikan pada Tabel 5.

Tabel 5. Pembagian Dataset Pelatihan dan Pengujian

Subset	Jumlah Citra	Keterangan
Training	80.000	Melatih Model
Validation	20.000	Memantau performa selama pelatihan
Testing	20.000	Evaluasi akhir model
Total	120.000	

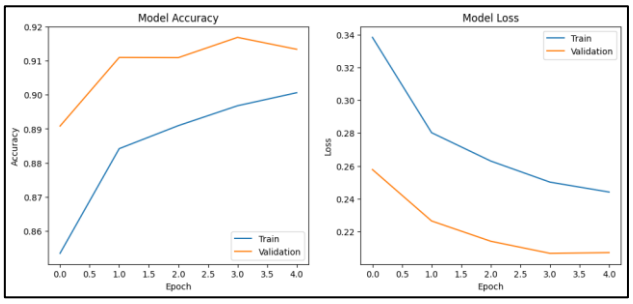
C. *Klasifikasi dengan Algoritma CNN EfficientNetB0*

Pada tahap ini dilakukan klasifikasi menggunakan EfficientNetB0 dengan pendekatan *transfer learning*. *Backbone* di-*freeze* dan ditambahkan *classifier head* baru. Parameter lengkap model disajikan pada Tabel 6.

Tabel 6. Parameter Pelatihan Model EfficientNetB0

Komponen	Detail
Backbone	EfficientNetB0 (pre-trained ImageNet, frozen)
Input Size	224 × 224 × 3
Layer Tambahan	GAP(Global Avarage Pooling) → Dense(512, ReLU) → Dropout(0,4) → Dense(256, ReLU) → Dropout(0,4) → Dense(1, Sigmoid)
Optimizer	AdamW
Learning Rate	0.001
Weight Decay	0.0001
Loss Function	Binary Crossentropy
Batch Size	64
Epochs	5
Callback	EarlyStopping, ModelCheckpoint
Trainable Parameters	787,457
Waktu Training	± 1 jam 12 menit

Selama pelatihan 5 *epoch*, model menunjukkan peningkatan performa yang bertahap. Akurasi *training* meningkat dari sekitar 85% pada *epoch* pertama menjadi 90,06% pada *epoch* kelima, sedangkan akurasi *validation* berada pada kisaran 89% hingga 91,33%. Nilai *loss training* menurun secara konsisten dari sekitar 0,34 menjadi 0,2441, dan *loss validation* turun hingga sekitar 0,2072. Kurva akurasi dan *loss* per *epoch* disajikan pada Gambar 3.



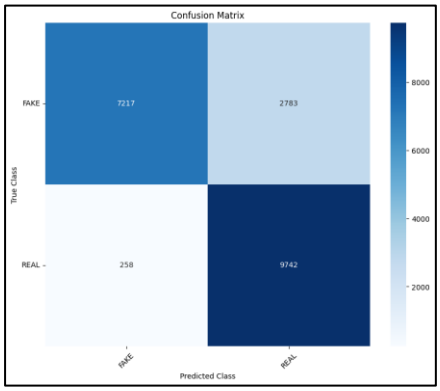
Gambar 3. Kurva Akurasi dan Loss Pelatihan Model EfficientNetB0 per Epoch

Evaluasi akhir pada data *testing* sebanyak 20.000 citra menghasilkan *classification report* yang ditampilkan pada Tabel 7.

Tabel 7. Classification Report Model EfficientNetB0

Kelas	Precision	Recall	F1-Score	Support
FAKE	0.9655	0.7217	0.826	10,000
REAL	0.7778	0.9742	0.865	10,000
Accuracy			0.848	20,000
Macro avg	0.8716	0.848	0.8455	20,000
Weighted avg	0.8716	0.848	0.8455	20,000

Confusion matrix model EfficientNetB0 disajikan pada Gambar 4.



Gambar 4. Visualisasi Confusion Matrix Model EfficientNetB0

Berdasarkan Tabel 7 dan Gambar 4, model berhasil mengklasifikasikan 9.742 citra *real* dengan benar (TP) dan 7.217 citra *fake* dengan benar (TN). Namun, terdapat 2.783 citra *fake* yang salah diklasifikasikan sebagai *real* (FP) dan hanya 258 citra *real* yang salah diklasifikasikan sebagai *fake* (FN).

Pola ini tercermin pada *classification report* di mana kelas REAL memiliki *recall* sangat tinggi (0,9742) namun *precision* lebih rendah (0,7778). Sebaliknya, kelas FAKE memiliki *precision* sangat tinggi (0,9655) namun *recall* yang lebih rendah (0,7217). Nilai *precision* pada kelas FAKE sebesar 0,9655 menunjukkan bahwa dari seluruh citra yang diprediksi sebagai *fake*, 96,55% benar-benar merupakan citra *fake*. Namun nilai *recall* 0,7217 mengindikasikan bahwa hanya 72,17% dari seluruh citra *fake* yang berhasil terdeteksi. Model cenderung bias terhadap kelas *real* dan masih kesulitan mengenali sebagian citra *fake*. Tingginya *false positive* sebesar 2.783 menjadi limitasi krusial dalam konteks deteksi konten manipulatif [16].

Secara keseluruhan model CNN EfficientNetB0 memperoleh akurasi *testing* 84,79% dengan 16.959 dari 20.000 citra terklasifikasi benar. Nilai *macro average* F1-*score* sebesar 0,8455 mengindikasikan ketidakseimbangan performa antar kelas, terutama terlihat dari perbedaan nilai *recall* yang signifikan.

D. *Klasifikasi dengan Vision Transformer*

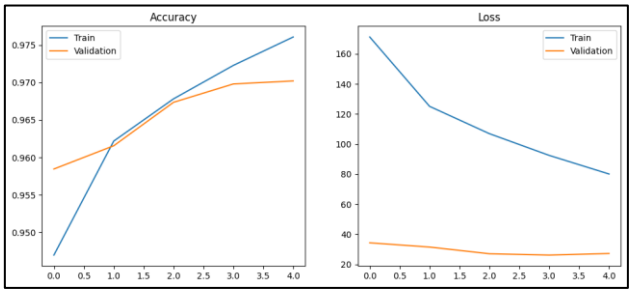
Pada tahap ini dilakukan klasifikasi menggunakan ViT-Base Patch16-224 dari Hugging Face Transformers yang diimplementasikan dengan PyTorch. *Backbone* di-*freeze* dan ditambahkan *classifier head* yang setara dengan EfficientNetB0. Parameter lengkap model disajikan pada Tabel 8.

Tabel 8. Parameter Pelatihan Model Vision Transformer

Komponen	Detail
Backbone	ViT-Base Patch16-224 (Hugging Face, frozen)
Input Size	224 × 224 × 3
Layer Tambahan	Linear(768 → 512, ReLU) → Dropout(0,4) → Linear(512 → 256, ReLU) → Dropout(0,4) → Linear(256 → 1, Sigmoid)
Optimizer	AdamW
Learning Rate	0.001
Weight	0.0001

Decay	
Loss Function	BCELoss
Batch Size	64
Epochs	5
Callback	EarlyStopping, ModelCheckpoint (scratch)
Trainable Parameters	525,313
Waktu Training	± 14 jam 18 menit

Vision Transformer menunjukkan karakteristik pelatihan yang sangat berbeda dari EfficientNetB0. Konvergensi terjadi jauh lebih cepat dengan akurasi tinggi sejak *epoch* pertama. Progres pelatihan selama 5 *epoch* disajikan pada Gambar 5.



Gambar 5. Kurva Akurasi dan Loss Pelatihan Model Vision Transformer per Epoch

Berdasarkan Gambar 5, pada *epoch* pertama ViT sudah mencapai akurasi *training* 94,70% dan *validation* 95,85%, melampaui akurasi terbaik EfficientNetB0. Peningkatan berlanjut secara konsisten hingga *epoch* kelima dengan *training accuracy* 97,61% dan *validation accuracy* 97,02%. Selisih antara akurasi *training* dan *validation* hanya sekitar 0,59% pada *epoch* terakhir, menunjukkan kemampuan generalisasi yang sangat baik tanpa indikasi *overfitting* yang signifikan [17].

Pada grafik *loss*, nilai *training loss* menurun tajam dari sekitar 171 pada *epoch* pertama menjadi sekitar 80 pada *epoch* kelima. *Validation loss* juga menurun stabil dari sekitar 34 menjadi sekitar 27. Tren ini menegaskan bahwa model telah mencapai konvergensi dengan baik selama pelatihan berlangsung.

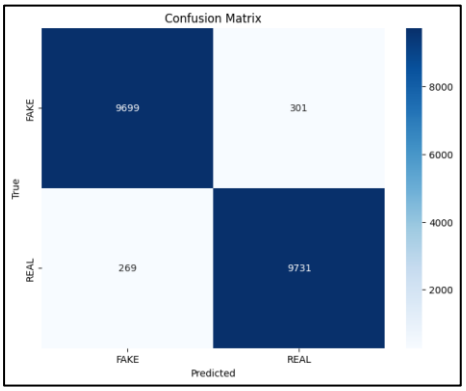
Evaluasi akhir pada data *testing* menghasilkan *classification report* pada Tabel 9.

Tabel 9. Classification Report Model Vision Transformer

Kelas	Precision	Recall	F1-Score	Support
FAKE	0.973	0.9699	0.9715	10,000
REAL	0.97	0.9731	0.9715	10,000
Accuracy			0.9715	20,000
Macro avg	0.9715	0.9715	0.9715	20,000
Weighted avg	0.9715	0.9715	0.9715	20,000

Lalu, *Confusion matrix* Model Vision Transformer disajikan pada Gambar 6.

Berdasarkan Tabel 9 dan Gambar 6, model ViT berhasil mengklasifikasikan 9.699 citra *fake* dengan benar (TN) dan 9.731 citra *real* dengan benar (TP). Kesalahan klasifikasi relatif kecil: hanya 301 citra *fake* yang salah diprediksi sebagai *real* (FP) dan 269 citra *real* salah diprediksi sebagai *fake* (FN).



Gambar 6. Visualisasi Confusion Matrix Model Vision Transformer

Pada kelas FAKE, *precision* sebesar 0,9730 menunjukkan 97,30% dari citra yang diprediksi *fake* benar-benar merupakan citra *fake*. Nilai *recall* 0,9699 menunjukkan 96,99% dari seluruh citra *fake* berhasil terdeteksi dengan benar — jauh melampaui *recall* EfficientNetB0 yang hanya 72,17%. F1-*score* sebesar 0,9715 menunjukkan keseimbangan yang sangat baik antara *precision* dan *recall* pada kelas *fake*.

Pada kelas REAL, *precision* 0,9700 dan *recall* 0,9731 menunjukkan performa yang hampir identik dengan kelas FAKE. Nilai *precision*, *recall*, dan F1-*score* yang hampir sama pada kedua kelas mengindikasikan bahwa model tidak menunjukkan bias terhadap salah satu kategori. Secara keseluruhan, model ViT mencapai akurasi *testing* 97,15% dengan 19.430 dari 20.000 citra terklasifikasi benar, dengan *macro average* F1-*score* 0,9715.

E. Analisis Perbandingan CNN EfficientNetB0 dan Vision Transformer

Perbandingan komprehensif kinerja kedua model disajikan pada Tabel 10.

Kriteria	CNN (EfficientNetB0)	Vision Transformer (ViT)
Train Accuracy	90.06%	97.61%
Validation Accuracy	91.33%	97.02%
Train Loss	0.2441	0.000103
Validation Loss	0.2072	0.00135
Test Accuracy	84.79%	97.15%
Data Test Benar	16.959 / 20.000	19.430 / 20.000
Precision (FAKE)	96.55%	97.30%
Recall (FAKE)	72.17%	96.99%
F1-score (FAKE)	82.60%	97.15%
Precision (REAL)	77.78%	97.00%
Recall (REAL)	97.42%	97.31%
F1-score (REAL)	86.50%	97.15%
Macro Avg F1-score	84.55%	97.15%
False Positive (Fake→Real)	2.783 citra	301 citra
False Negative (Real→Fake)	258 citra	269 citra
Konvergensi	Bertahap	Cepat dan stabil

Kriteria	CNN (EfficientNetB0)	Vision Transformer (ViT)
Training	(~85%→90%)	(>94% epoch 1)
Karakteristik Model	Fitur lokal (konvolusi)	Fitur global (self-attention)
Waktu Training (5 epoch)	±1 jam 12 menit	±14 jam 18 menit
Keseimbangan Model	Kurang seimbang (bias ke REAL)	Seimbang antar kelas
Trainable Parameters	787,457	525,313

Berdasarkan Tabel 12, terdapat perbedaan yang signifikan antara EfficientNetB0 dan ViT-Base Patch16-224 pada hampir seluruh metrik evaluasi. Vision Transformer memperoleh akurasi testing sebesar 97,15%, lebih tinggi dibandingkan EfficientNetB0 sebesar 84,79%. Selain itu, Vision Transformer menunjukkan kemampuan deteksi citra fake yang lebih baik dengan recall kelas FAKE sebesar 96,99%, sedangkan EfficientNetB0 hanya mencapai 72,17%.

Dari sisi keseimbangan performa, Vision Transformer menghasilkan precision, recall, dan F1-score yang hampir sama pada kedua kelas sehingga menunjukkan kemampuan generalisasi yang lebih baik. Jumlah false positive juga jauh lebih rendah dibandingkan EfficientNetB0. Namun, EfficientNetB0 memiliki keunggulan dalam efisiensi komputasi dengan waktu pelatihan yang lebih cepat. Secara keseluruhan, Vision Transformer lebih unggul dalam kualitas klasifikasi, sedangkan EfficientNetB0 lebih efisien dari sisi komputasi.

IV. KESIMPULAN

Berdasarkan hasil eksperimen pada dataset CIFAKE, EfficientNetB0 dan ViT-Base Patch16-224 menunjukkan karakteristik yang berbeda. EfficientNetB0 memperoleh akurasi testing 84,79% dengan waktu pelatihan yang efisien (±1 jam 12 menit), namun masih lemah dalam mendeteksi citra fake dengan recall kelas FAKE sebesar 72,17% dan false positive yang tinggi. Sebaliknya, Vision Transformer memberikan performa yang jauh lebih baik dengan akurasi 97,15% serta precision, recall, dan F1-score yang seimbang pada kedua kelas, meskipun membutuhkan waktu pelatihan lebih lama (±14 jam 18 menit).

Secara keseluruhan, Vision Transformer lebih unggul dalam kualitas klasifikasi, sedangkan EfficientNetB0 lebih efisien secara komputasi. Penelitian selanjutnya dapat mengembangkan arsitektur hybrid CNN–Transformer, memperluas dataset, serta menerapkan fine-tuning dan hyperparameter tuning untuk meningkatkan performa model.

DAFTAR PUSTAKA

[1] B. Liu, B. Liu, T. Zhu, and M. Ding, "A Review of Deepfake and Its Detection: From Generative Adversarial Networks to Diffusion Models," 2025, *John Wiley and Sons Inc*. doi: 10.1155/int/9987535.

[2] Dhruv Bhatada, Nimesh Gujari, Priyanka Pandey, Yashvant Chhapwale, and Dr. Yogita Shelar, "AI-Generated-Image Detection Using Deep Learning Techniques," *International Journal of Advanced Research in Science, Communication and Technology*, pp. 292–302, Apr. 2025, doi: 10.48175/ijarsct-25140.

[3] S. Aribe, "A Hybrid Deep Learning and Forensic Approach for Robust Deepfake Detection," Oct. 2025. [Online]. Available: www.ijacsa.thesai.org

[4] D. A. Ramadhan and D. Ramadhani, "Classification of Riau Batik Motifs Using the Convolutional Neural Network (CNN) Algorithm," *International Journal of Electrical, Energy and Power System Engineering*, vol. 7, no. 3, pp. 201–211, Nov. 2024, doi: 10.31258/ijeepse.7.3.201-211.

[5] L. Wahyuni and D. Ramadhani, "CNN-Based Eucalyptus Disease Classification Using MobileNet," *International Journal of Electrical, Energy and Power System Engineering*, vol. 8, no. 3, pp. 293–304, Oct. 2025, doi: 10.31258/ijeepse.8.3.293-304.

[6] H. A. Shah, F. Saeed, S. Yun, J. H. Park, A. Paul, and J. M. Kang, "A Robust Approach for Brain Tumor Detection in Magnetic Resonance Images Using Finetuned EfficientNet," *IEEE Access*, vol. 10, pp. 65426–65438, 2022, doi: 10.1109/ACCESS.2022.3184113.

[7] R. Hernández-López, C. M. Travieso-González, and N. I. Ajali-Hernández, "Sky Image Classification Based on Transfer Learning Approaches," *Sensors*, vol. 24, no. 12, Jun. 2024, doi: 10.3390/s24123726.

[8] Y. Wang, Y. Deng, Y. Zheng, P. Chattopadhyay, and L. Wang, "Vision Transformers for Image Classification: A Comparative Survey," Jan. 01, 2025, *Multidisciplinary Digital Publishing Institute (MDPI)*. doi: 10.3390/technologies13010032.

[9] D. A. Coccomini, R. Caldelli, F. Falchi, C. Gennaro, and G. Amato, "Cross-Forgery Analysis of Vision Transformers and CNNs for Deepfake Image Detection," in *MAD 2022 - Proceedings of the 1st International Workshop on Multimedia AI against Disinformation*, Association for Computing Machinery, Inc, Jun. 2022, pp. 52–58. doi: 10.1145/3512732.3533582.

[10] J. J. Bird and A. Lotfi, "CIFAKE: Image Classification and Explainable Identification of AI-Generated Synthetic Images," *IEEE Access*, vol. 12, pp. 15642–15650, 2024, doi: 10.1109/ACCESS.2024.3356122.

[11] S. Kumar, P. Asiamah, O. Jolaoso, and U. Esiowu, "Enhancing Image Classification with Augmentation: Data Augmentation Techniques for Improved Image Classification," Feb. 2025, [Online]. Available: http://arxiv.org/abs/2502.18691

[12] K. Alomar, H. I. Aysel, and X. Cai, "Data Augmentation in Classification and Segmentation: A Survey and New Strategies," *J. Imaging*, vol. 9, no. 2, Feb. 2023, doi: 10.3390/jimaging9020046.

[13] B. Ghita, I. Kuzminykh, A. Usama, T. Bakhshi, and J. Marchang, "Deepfake Image Detection using Vision Transformer Models," IEEE, Jun. 2024. [Online]. Available: http://shura.shu.ac.uk/information.html

[14] J. Ha, A. El Azzaoui, and J. H. Park, "FL-TENB4: A Federated-Learning-Enhanced Tiny EfficientNetB4-Lite Approach for Deepfake Detection in CCTV Environments," *Sensors*, vol. 25, no. 3, Feb. 2025, doi: 10.3390/s25030788.

[15] M. Sharafudeen, A. J, and V. Chandra S. S, "Leveraging Vision Attention Transformers for Detection of Artificially Synthesized Dermoscopic Lesion Deepfakes Using Derm-CGAN," *Diagnostics*, vol. 13, no. 5, Mar. 2023, doi: 10.3390/diagnostics13050825.

[16] B. G. Deepa, C. K. Lokesh, D. Umamaheswari, B. Ayshwarya, P. V. Yethish, and K. P. Suhaas, "An enhanced deep learning framework for DeepFake detection using EfficientNet-B3 comparative evaluation of deep and machine learning techniques," *Discover Computing*, vol. 29, no. 1, Jan. 2026, doi: 10.1007/s10791-025-09890-x.

[17] D. W. Deressa, H. Mareen, P. Lambert, S. Atnafu, Z. Akhtar, and G. Van Wallendael, "GenConViT: Deepfake Video Detection Using Generative Convolutional Vision Transformer," Mar. 2025, [Online]. Available: http://arxiv.org/abs/2307.07036.