

# Laporan Project - Pembelajaran Mesin

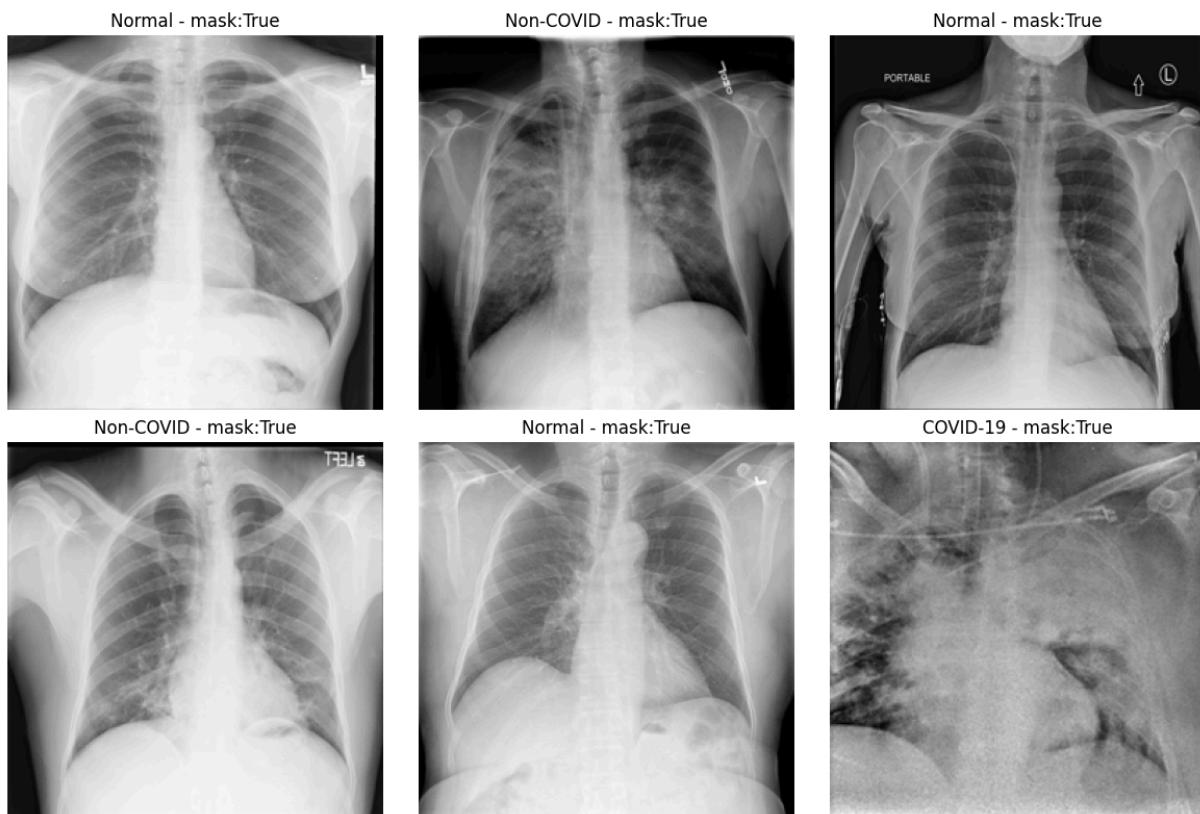
Anggota Kelompok :

202210370311060 - Muhammad Hisyam Kamil  
[hisyamkamil99@webmail.umm.ac.id](mailto:hisyamkamil99@webmail.umm.ac.id) - +6289680855665

202210370311449 - Elga Putri Tri Farma  
[elgafarma@webmail.umm.ac.id](mailto:elgafarma@webmail.umm.ac.id) - +62895372762346

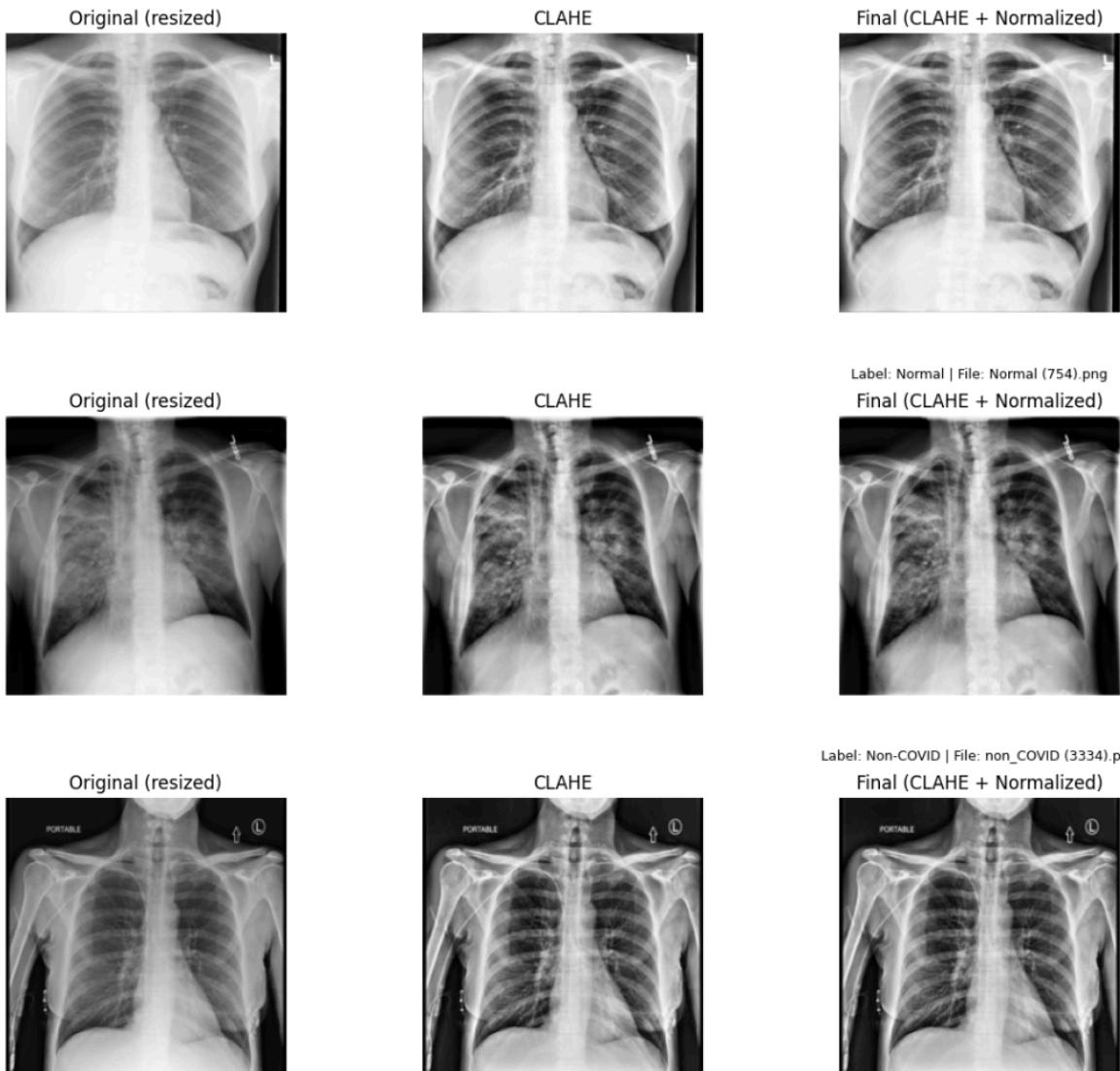
## Tahap 1: Preprocessing Data

Dataset yang digunakan adalah COVID-QU-Ex. Total terdapat 5.826 citra yang terbagi ke dalam tiga kelas dengan proporsi COVID-19 sebanyak 2.913 gambar, Non-COVID 1.457 gambar, dan Normal 1.456 gambar. Struktur folder sudah dipisah Train, Val, dan Test. Semua artefak dan keluaran eksperimen disimpan rapi di direktori artifacts/ sehingga proses ulang maupun peninjauan kembali dapat dilakukan tanpa kesulitan. Seluruh percobaan dijalankan dengan seed tetap (RANDOM\_STATE=42) dan pembagian data stratified agar proporsi kelas terjaga di setiap subset.



Gbr. 1.1. Sampel Citra Chest X-ray Dataset COVID-QU-Ex

Langkah pertama adalah membuat semua gambar berada pada kondisi yang seragam dan lebih “bersih” untuk dianalisis. Seluruh citra dibaca sebagai grayscale, diubah ukurannya menjadi  $224 \times 224$  piksel, lalu ditingkatkan kontrasnya menggunakan CLAHE dengan pengaturan clip limit 2.0 dan tile grid  $8 \times 8$ . Pilihan CLAHE didasari kebutuhan untuk menonjolkan struktur paru—seperti garis interstisial, batas lobus, atau area opasitas—tanpa memunculkan artefak berlebihan. Setelah itu, nilai piksel dinormalisasi ke rentang  $[0, 1]$  agar skala input konsisten di seluruh gambar.



**Gbr. 1.2. Perbandingan Citra Resized, dengan CLAHE, dan CLAHE+Normalized**

Semua operasi ini diterapkan persis sama pada data latih, validasi, dan uji, serta akan digunakan kembali ketika melakukan inferensi pada satu gambar. Di tahap ini tidak ada satupun pemanggilan fungsi pembaca mask ataupun operasi yang memanfaatkan mask; pipeline

benar-benar murni gambar rontgen saja. Hasil pra-proses disimpan sebagai berkas NumPy processed\_infection.npz yang memuat array citra berukuran (5826, 224, 224), label numerik (COVID-19→0, Non-COVID→1, Normal→2), dan daftar path sumber file. Pemeriksaan cepat memastikan tidak ada gambar yang menjadi gelap total setelah diproses. Untuk memastikan kualitas, kami juga menghasilkan grid contoh “Original → CLAHE → Final” sehingga perubahan visual bisa dilihat dengan jelas.

**Tabel 1.1: Informasi fitur yang ada pada dataset**

Label	Jumlah Citra	Persentase
COVID-19	2.913	50.0%
Non-COVID	1.457	25.0%
Normal	1.456	25.0%
<b>Total</b>	<b>5.826</b>	<b>100%</b>

Pembagian data dilakukan dengan stratified split menjadi 70% latih, 15% validasi, dan 15% uji. Hasilnya adalah 4.078 gambar pada train, 874 pada validation, dan 874 pada test. Indeks pembagian disimpan dalam splits.npz agar pembagian yang sama bisa dipakai berulang kali tanpa perbedaan.

**Tabel 1.2: Informasi pembagian dataset**

Subset	Jumlah Citra	Persentase	Keterangan
Train	4.078	70%	Digunakan untuk melatih model
Validation	874	15%	Digunakan untuk tuning parameter dan mencegah overfitting
Test	874	15%	Digunakan untuk evaluasi akhir kinerja model
<b>Total</b>	<b>5.826</b>	<b>100%</b>	-

## **Tahap 2: Feature Engineering**

(Feature Extraction → Feature Selection → Feature Scaling)

Setelah gambar siap, masuk ke tahap feature engineering yang menjadi inti dari pipeline klasik ini. Tujuannya adalah mengubah gambar yang sudah seragam tadi menjadi representasi numerik yang kaya informasi, tetapi tetap ringkas dan stabil untuk dipakai oleh model.

Bagian pertama adalah feature extraction menggunakan HOG (Histogram of Oriented Gradients). Dengan parameter orientasi 9, ukuran sel  $16 \times 16$  piksel, dan blok  $2 \times 2$  sel, HOG menangkap pola tepi dan tekstur yang muncul dari perbedaan intensitas pada jaringan paru. Hasil ekstraksi per gambar menghasilkan vektor sepanjang 6.084 fitur. Seluruh matriks fitur ini disimpan ke hog features.npy.

**Tabel 2.1. Konfigurasi HOG**

Parameter	Nilai	Alasan Singkat
orientations	9	Granularitas arah tepi yang umum dipakai
pixels_per_cell	(16, 16)	Menangkap tekstur paru pada skala lokal
cells_per_block	(2, 2)	Normalisasi gradien yang stabil
feature_vector	True	Vektor 1D siap ke tahap seleksi & skala
<b>Dimensi HOG</b>	<b>6.084</b>	Setelah diratakan per gambar

Jumlah fitur tersebut cukup besar, sehingga langkah berikutnya adalah feature selection dengan SelectKBest menggunakan skor ANOVA (f\_classif). Seleksi ini murni dipelajari dari data latih saja. Untuk menentukan berapa banyak fitur yang ideal, kami menguji beberapa kandidat nilai k (256 hingga 4.096) dan menilai kinerjanya pada validation set menggunakan macro-F1. Kinerja meningkat konsisten sampai k=4.096 dan berhenti membaik setelahnya; oleh karena itu kami menetapkan k=4.096 sebagai pilihan akhir. Selektor yang sudah terlatih disimpan sebagai feature\_selector.joblib.

**Tabel 2.2. Uji nilai k pada SelectKBest (val macro-F1)**

k	Macro-F1 (Val)
256	0,6843
512	0,6944
1.024	0,7317
2.048	0,7462
<b>4.096</b>	<b>0,7669</b>

Setelah fitur diseleksi, barulah dilakukan feature scaling menggunakan StandardScaler. Penskalaan ditempatkan setelah seleksi agar statistik mean dan deviasi standar dihitung persis pada subruang fitur yang betul-betul dipakai model. StandardScaler juga dilatih hanya pada data latih, kemudian dipakai untuk mentransformasi validation dan test. Artefaknya disimpan sebagai feature\_scaler.joblib. Dengan demikian, urutan rekayasa fitur selalu sama: HOG, lalu SelectKBest, baru StandardScaler. Urutan yang konsisten ini kami gunakan hingga ke tahap inferensi satu gambar.

**Tabel 2.3. Standardisasi fitur**

Komponen	Kebijakan Fit/Transform	Catatan
SelectKBest	Fit di Train → Transform Val/Test	k=4096, f_classif
StandardScaler	Fit di Train → Transform Val/Test	Skala setelah seleksi

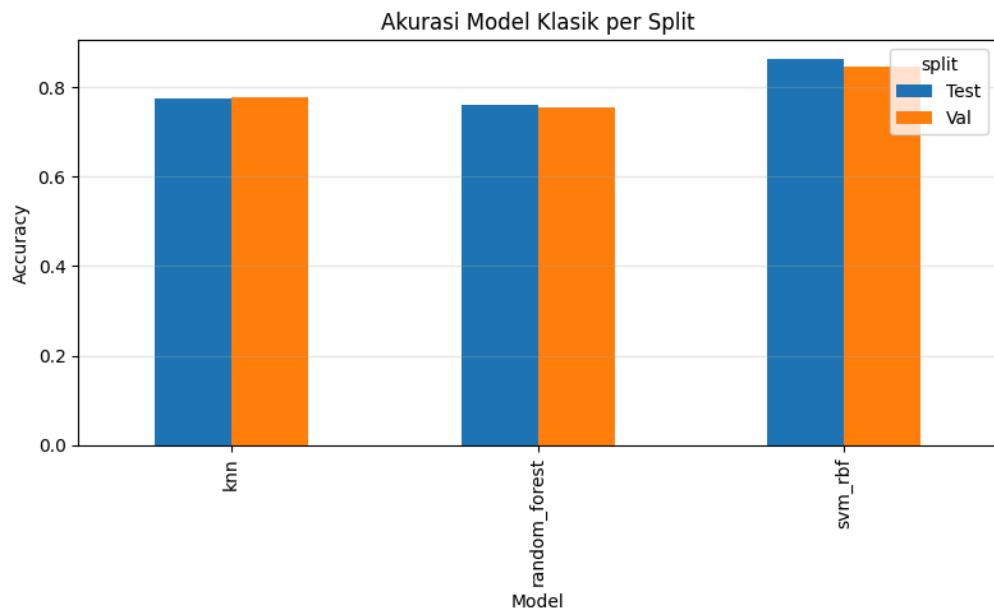
### Tahap 3: Pelatihan Model Klasik

Dengan fitur yang sudah siap, kami melatih tiga model klasik: SVM dengan kernel RBF, Random Forest, dan k-Nearest Neighbors. SVM RBF memakai C=10, gamma “scale”, class weight “balanced”, dan keluaran probabilitas diaktifkan. Random Forest kami atur dengan 300 pohon dan class weight balanced\_subsample. Sementara itu kNN memakai k=5 tetangga terdekat. Semua model dilatih pada data latih yang telah melalui pipeline HOG → seleksi → skala, kemudian dievaluasi di validation dan test, dan disimpan ke folder classic\_models untuk dipakai kembali tanpa harus melatih ulang.

**Tabel 3.1. Ringkasan hiperparameter model**

Model	Hiperparameter Utama
SVM RBF	C=10, gamma=“scale”, class_weight=“balanced”, probas=On
RandomForest	n_estimators=300, class_weight=“balanced_subsample”
kNN	n_neighbors=5

Evaluasi dilakukan dua kali: pertama pada validation set untuk memastikan pemilihan jumlah fitur hasil seleksi sudah masuk akal, lalu pada test set untuk pelaporan akhir. Metrik yang kami gunakan adalah akurasi dan macro-F1 agar kontribusi tiap kelas seimbang.



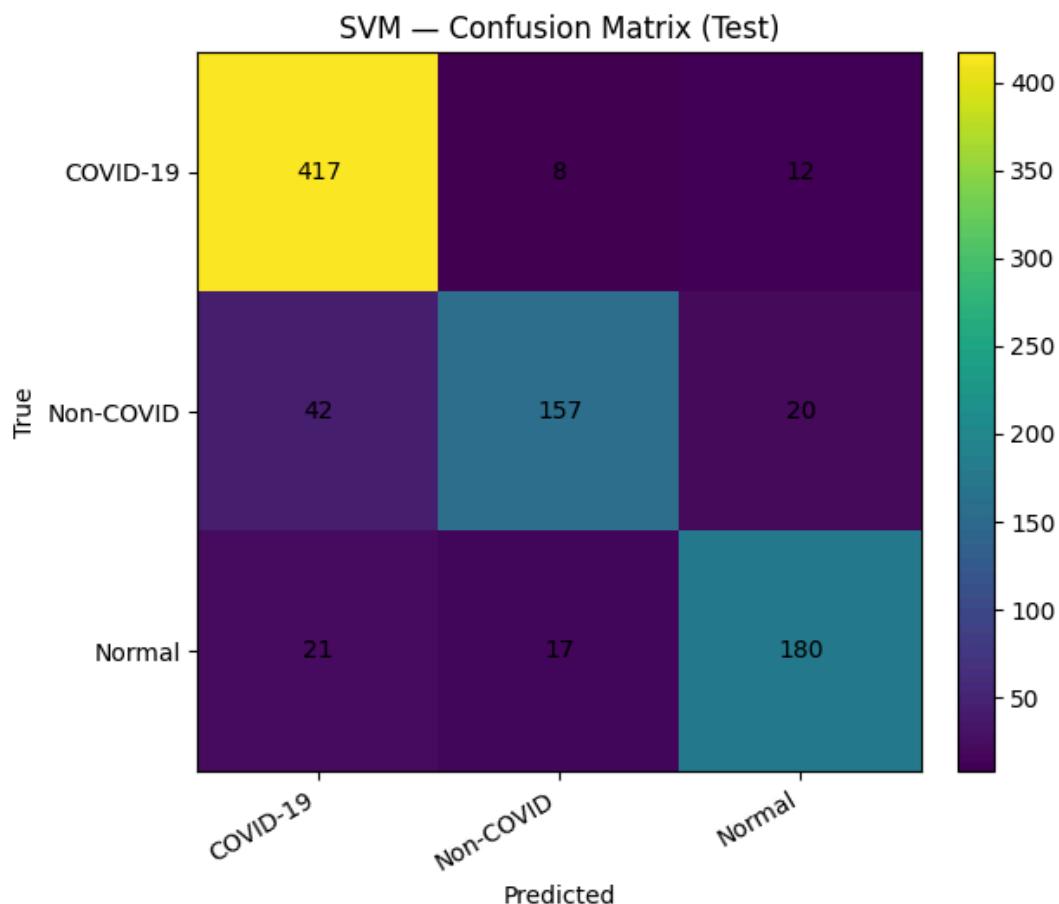
**Gbr. 3.1. Perbandingan Akurasi Model Klasik per Split**

Hasil akhirnya cukup jelas. SVM RBF tampil paling baik dan stabil: akurasi 84,78% dengan macro-F1 0,823 pada validation, dan meningkat di test menjadi akurasi 86,27% dengan macro-F1 0,843. Random Forest dan kNN berada di kisaran 75–78% akurasi dengan macro-F1 sekitar 0,70–0,74.

**Tabel 3.2. Kinerja model — Validation & Test**

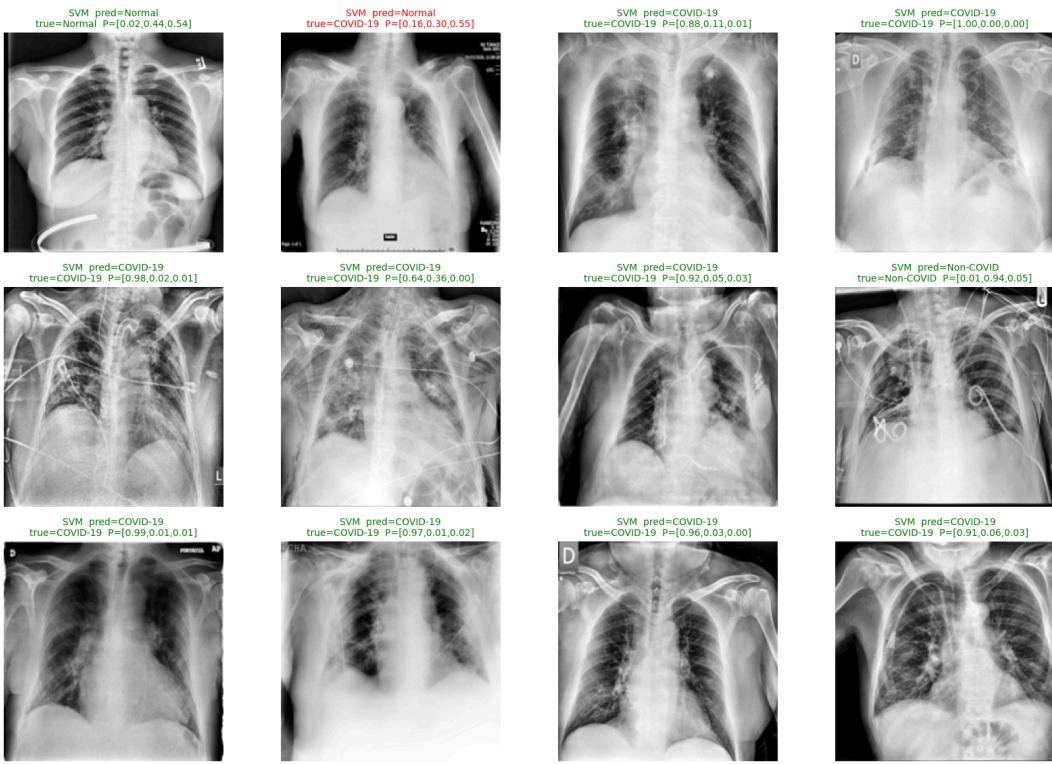
Model	Split	Akurasi	Precision (macro)	Recall (macro)	F1 (macro)
SVM RBF	Val	<b>0,8478</b>	0,8313	0,8178	<b>0,8234</b>
SVM RBF	Test	<b>0,8627</b>	0,8601	0,8323	<b>0,8432</b>
RandomForest	Val	0,7551	0,8050	0,6829	0,7044
RandomForest	Test	0,7609	0,8066	0,6966	0,7188
kNN (k=5)	Val	0,7769	0,7971	0,7195	0,7345
kNN (k=5)	Test	0,7757	0,7841	0,7225	0,7389

Confusion Matrix pada SVM memperlihatkan bahwa kelas COVID-19 terdeteksi sangat baik, dengan recall sekitar 95%.

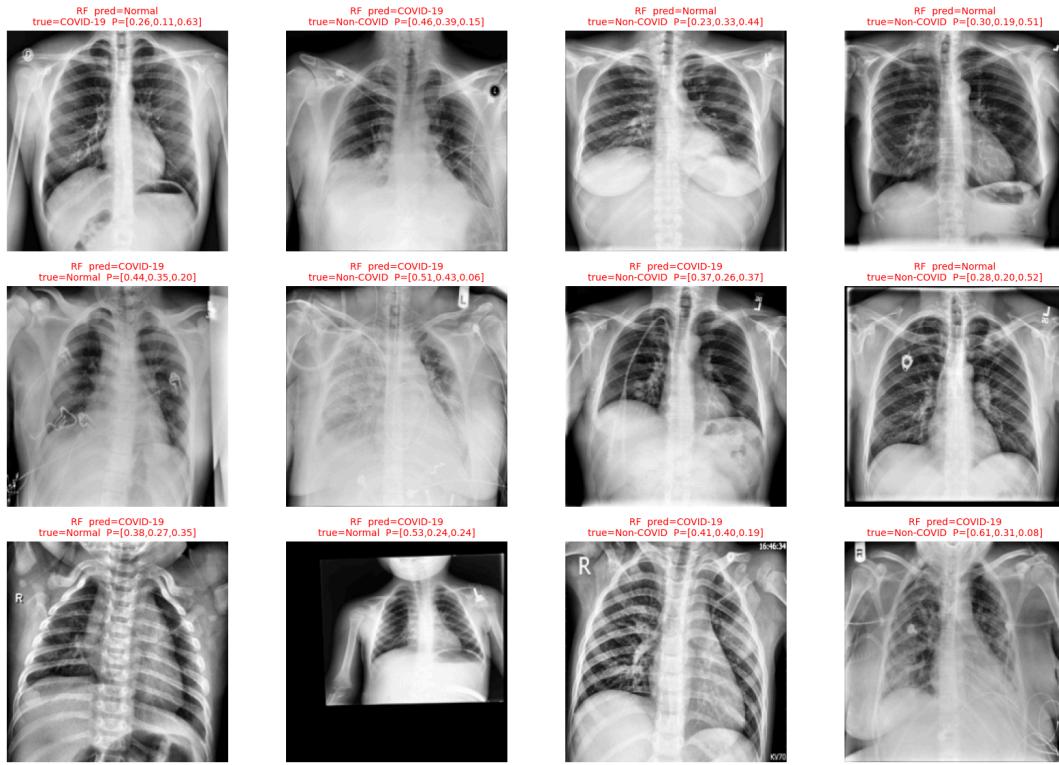


**Gbr. 3.2. Confusion Matrix SVM on Test Dataset**

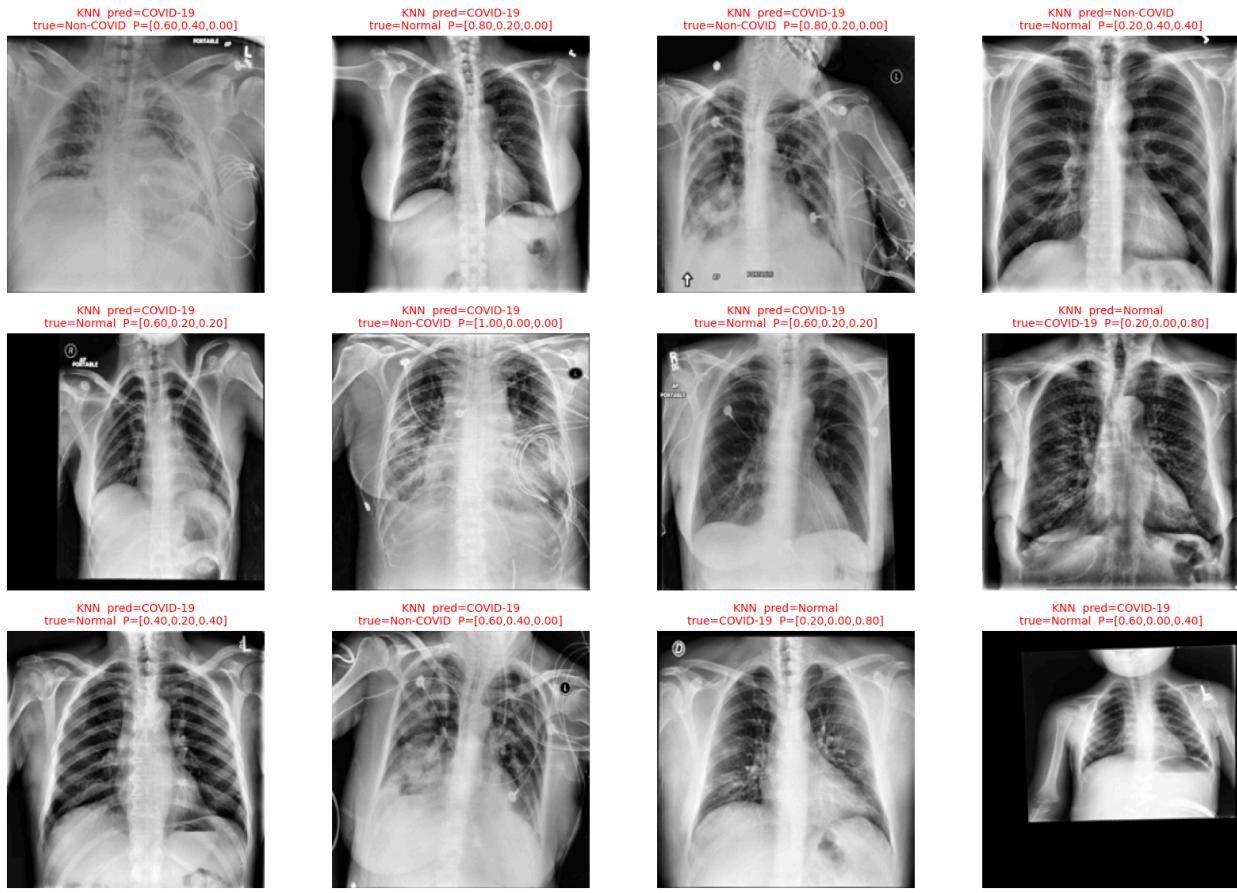
Kelas Non-COVID masih menjadi sumber kesalahan terbesar karena sebagian kasusnya didorong ke kelas COVID-19. Dari peninjauan visual, kesalahan seperti ini kerap terjadi pada citra yang menyertakan kabel, tube, atau artefak klinis lain yang memperkaya struktur tepi sehingga “terbaca” sebagai pola yang mirip lesi. Kelas Normal relatif lebih stabil; struktur paru yang bersih setelah CLAHE membuatnya lebih mudah dipisahkan dari dua kelas lainnya.



Gbr. 3.3. Sampel Hasil Prediksi Pada Model SVM menggunakan dataset Test



Gbr. 3.4. Sampel Hasil Prediksi Pada Model RandomForest menggunakan dataset Test



Gbr. 3.5. Sampel Hasil Prediksi Pada Model KNN menggunakan dataset Test

## Tahap 4: Analisis Kesalahan dan Interpretasi Model

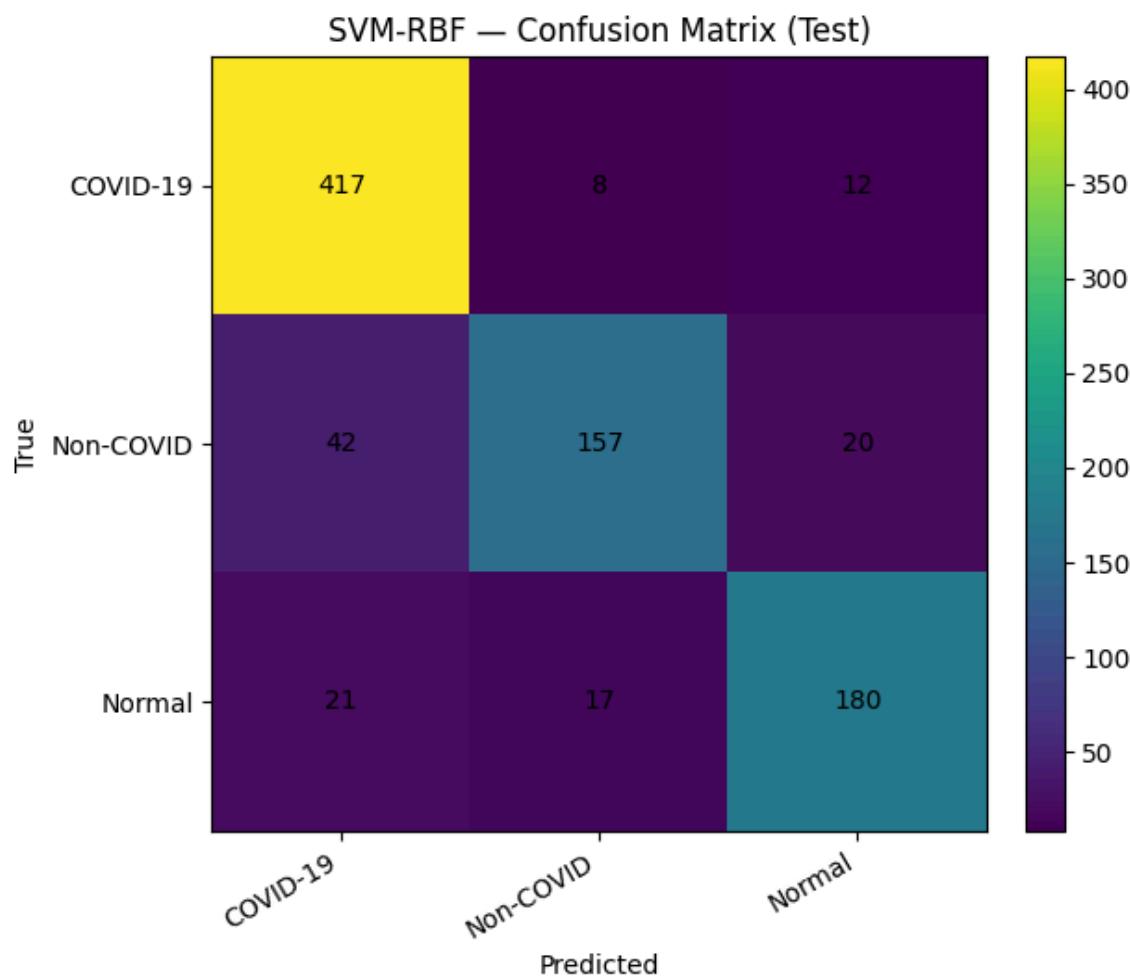
### 4.1 Tujuan dan Ruang Lingkup

Tahap ini bertujuan mengevaluasi lebih dalam perilaku model klasik (SVM RBF, Random Forest, dan kNN) bukan hanya dari metrik agregat, tetapi juga dari pola kesalahan, keyakinan prediksi, serta karakteristik fitur yang mendasari keputusan. Analisis dilakukan dalam empat lapisan:

1. Statistik dasar: confusion matrix dan classification report.
2. Interpretasi per model: PCA/t-SNE 2D, decision boundary, histogram confidence, serta feature importances.
3. Analisis kualitatif: visualisasi sampel benar/salah dan penelusuran tetangga terdekat.
4. Analisis lintas model + benchmark performa akhir.

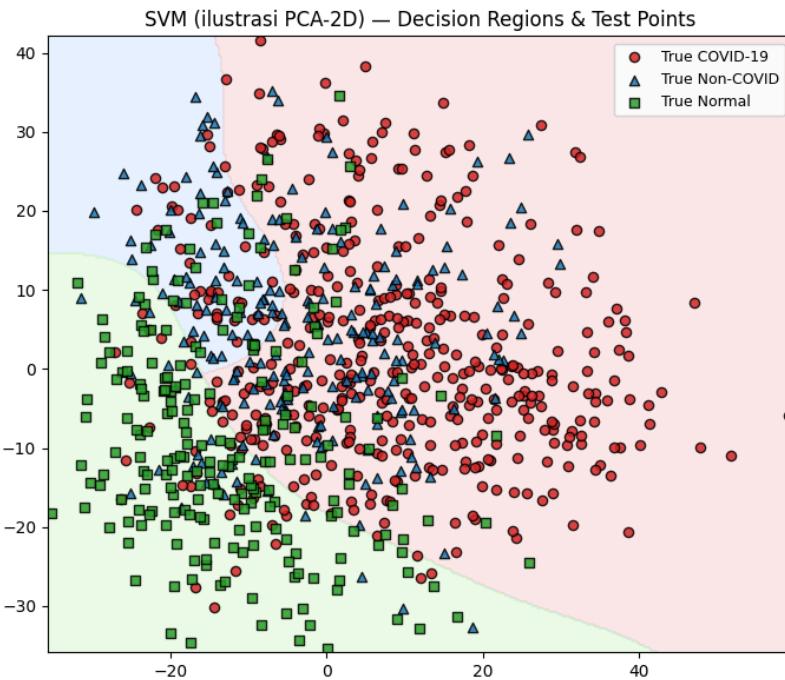
## 4.2 Analisis Model SVM-RBF

Model SVM RBF memberikan performa tertinggi (akurasi 86,27 % | macro-F1 0,843). **Confusion matrix** pada Gbr. 4.1 menunjukkan bahwa kelas COVID-19 memiliki recall  $\approx 95\%$ , sementara kesalahan terbesar muncul pada kelas Non-COVID yang sebagian besar terkласifikasi sebagai COVID-19. Hal ini menunjukkan model sangat sensitif terhadap pola infiltrasi paru namun kadang keliru terhadap artefak klinis.



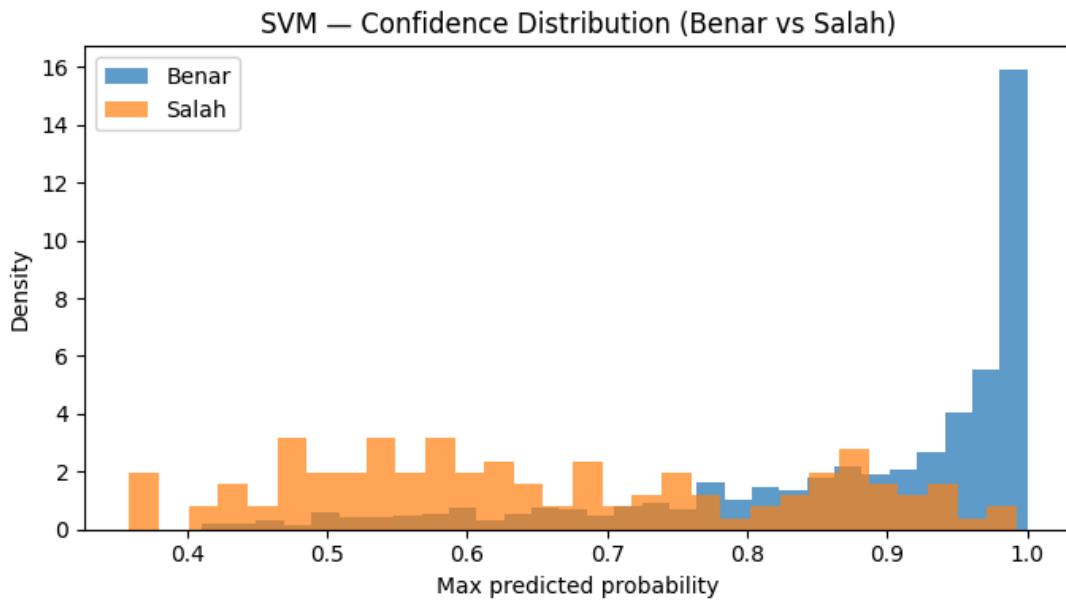
**Gbr. 4.1 Confusion Matrix SVM-RBF (Test)**

Visualisasi **decision boundary PCA-2D** (Gbr. 4.2) memperlihatkan pemisahan kelas yang relatif baik, tetapi masih terdapat tumpang tindih antara Non-COVID dan Normal pada area tengah. Ini menegaskan bahwa fitur HOG yang digunakan masih menghasilkan representasi beririsan pada jaringan paru yang relatif bersih.



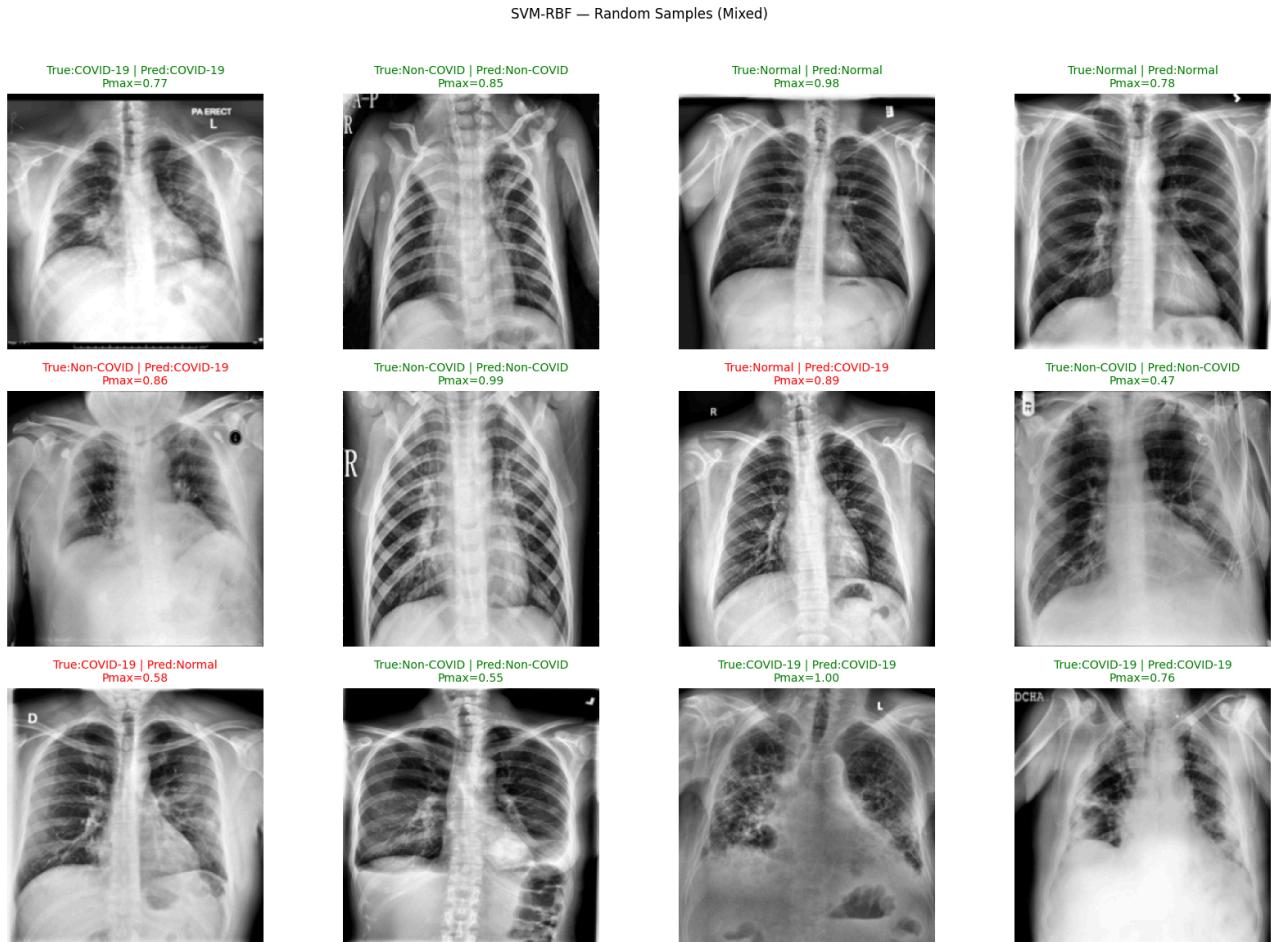
**Gbr. 4.2 SVM (ilustrasi PCA-2D) — Decision Regions & Test Points**

Distribusi **confidence score** (Gbr. 4.3) menunjukkan mayoritas prediksi benar memiliki probabilitas  $> 0,9$ , sedangkan prediksi salah tersebar di rentang 0,5–0,8. Artinya model cukup yakin saat benar, tetapi ragu saat salah — indikasi kalibrasi probabilitas yang baik.



**Gbr. 4.3 SVM — Confidence Distribution (Benar vs Salah)**

Sampel citra **prediksi acak dengan confidence** (Gbr. 4.4) memperlihatkan contoh benar (hijau) dan salah (merah). Kesalahan umumnya berasal dari citra Non-COVID yang memiliki kabel atau bayangan medis menyerupai opasitas.

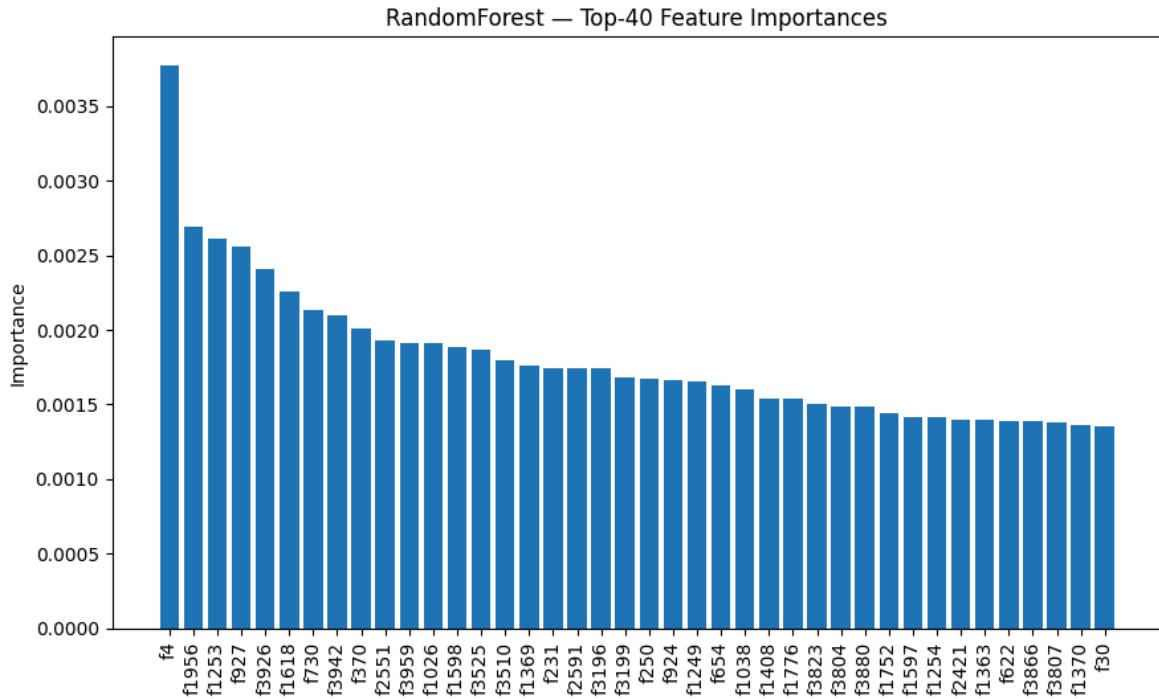


**Gbr. 4.4 Sampel Prediksi SVM RBF — Benar vs Salah dengan Confidence**

### 4.3 Analisis Model Random Forest

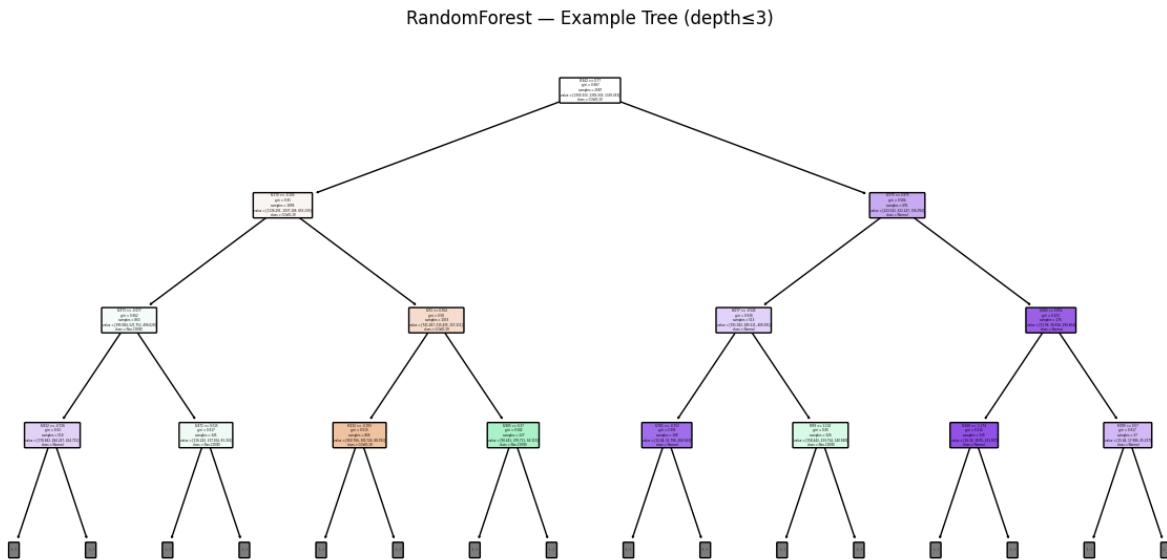
Random Forest mencapai akurasi 76 % | macro-F1 0,719. Model cenderung stabil tetapi tidak sekuat SVM karena keterbatasan dalam menangkap hubungan non-linier antar-fitur.

**Feature importances** (Gbr. 4.5) memperlihatkan kontribusi tertinggi berasal dari fitur HOG f4, f1956, dan f1253 yang dominan menggambarkan tekstur dan tepi pada area paru bawah — menandakan bagian tersebut paling berpengaruh dalam klasifikasi.



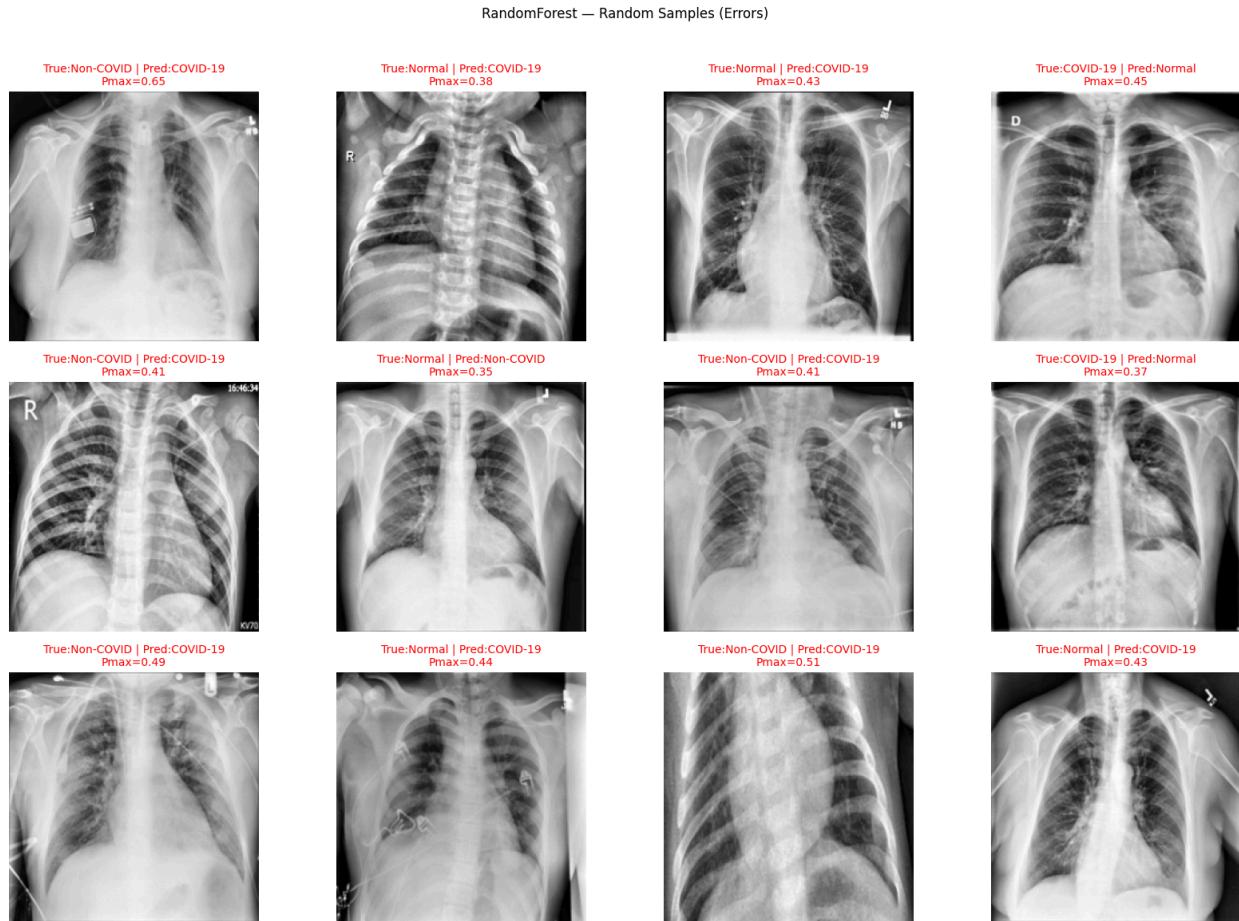
**Gbr. 4.5 RandomForest — Top-40 Feature Importances**

Untuk interpretasi struktur pohon, **visualisasi pohon dangkal** (Gbr. 4.6) menunjukkan pola keputusan yang mudah dibaca: simpul awal memisahkan COVID-19 vs lainnya berdasarkan fitur dominan, lalu percabangan berikutnya membedakan Normal dan Non-COVID berdasarkan kombinasi tekstur halus.



**Gbr. 4.6 RandomForest — Example Tree (depth ≤ 3)**

Contoh citra benar dan salah dari Random Forest (Gbr. 4.7) menegaskan bahwa model sering gagal saat citra terlalu homogen (low contrast) atau berisi noise sensor, menyebabkan split decision tidak cukup dalam untuk mengenali pola global.

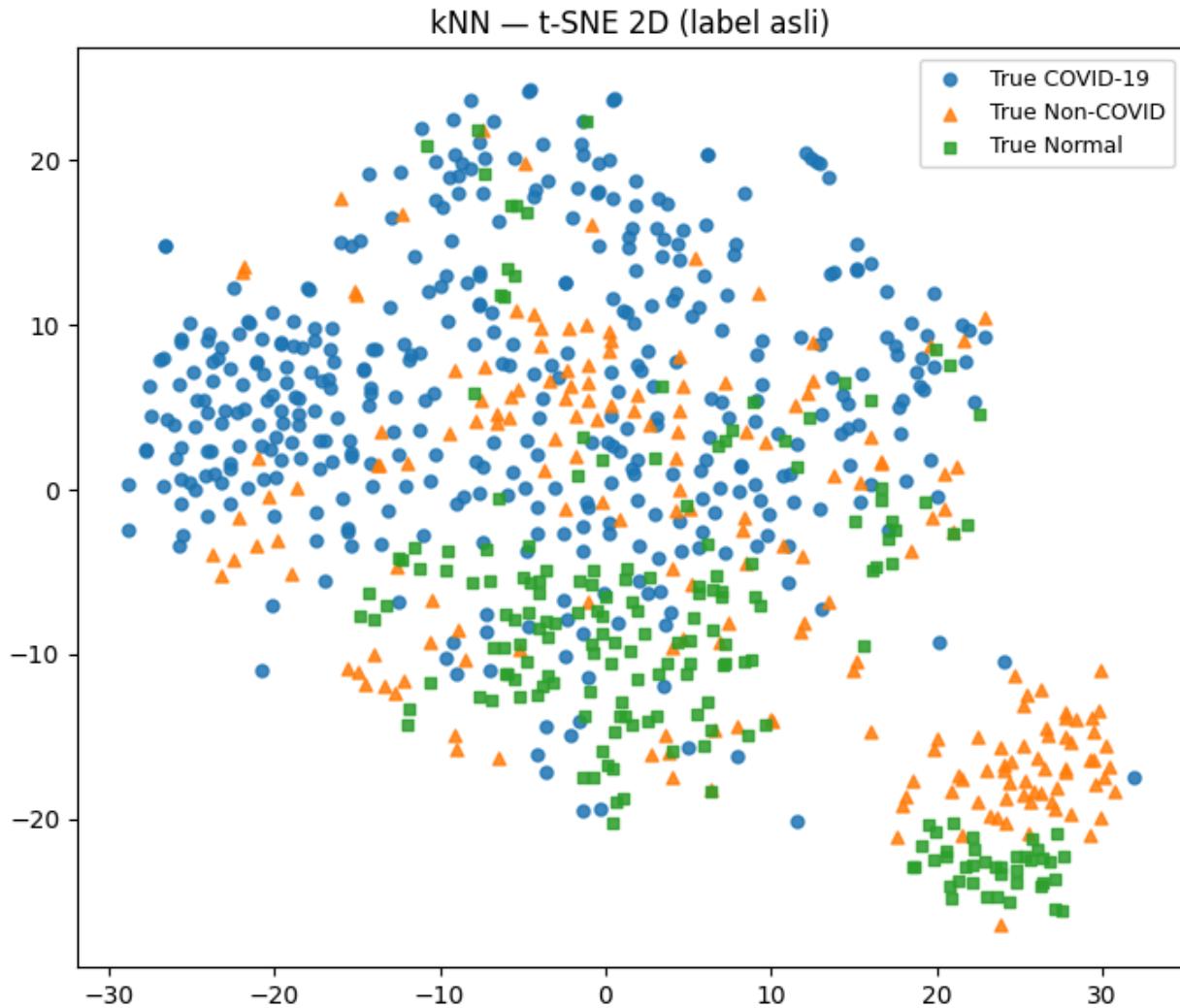


**Gbr. 4.7 Sampel Prediksi RandomForest (Test)**

#### 4.4 Analisis Model k-Nearest Neighbors (kNN)

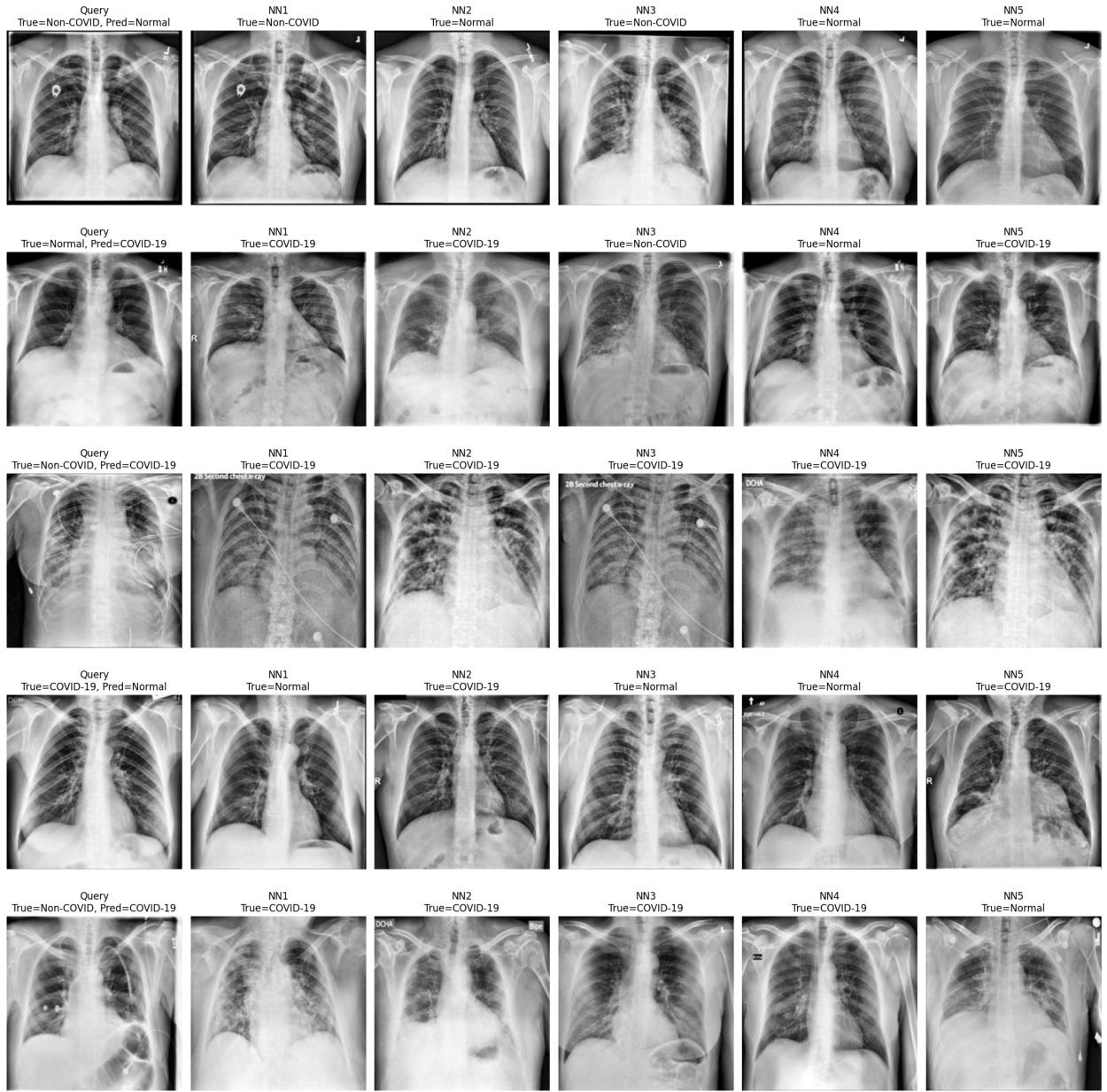
Model kNN dengan  $k = 5$  memiliki akurasi 77 % | macro-F1 0,739. Model ini sangat bergantung pada jarak fitur sehingga lebih rentan terhadap distribusi fitur yang padat.

Visualisasi t-SNE 2D (Gbr. 4.8) memperlihatkan pengelompokan COVID-19 yang cukup jelas, namun cluster Non-COVID dan Normal sering berdekatan. Hal ini menandakan jarak Euclidean pada fitur HOG belum sepenuhnya merepresentasikan variasi klinis.



**Gbr. 4.8 kNN — t-SNE 2D (label asli)**

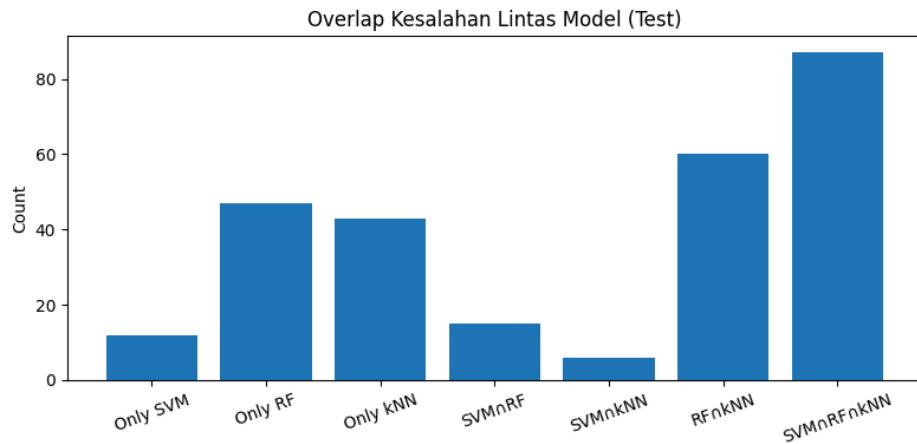
Analisis **5 tetangga terdekat** pada kasus salah (Gbr. 4.9) mengungkap bahwa beberapa citra salah diklasifikasi karena tetangga terdekat berasal dari kelas yang salah tetapi memiliki tekstur paru serupa — misal, COVID-19 dengan Non-COVID tanpa infiltrasi.



**Gbr. 4.9 kNN — Query vs 5 Nearest Neighbors (pada sampel salah)**

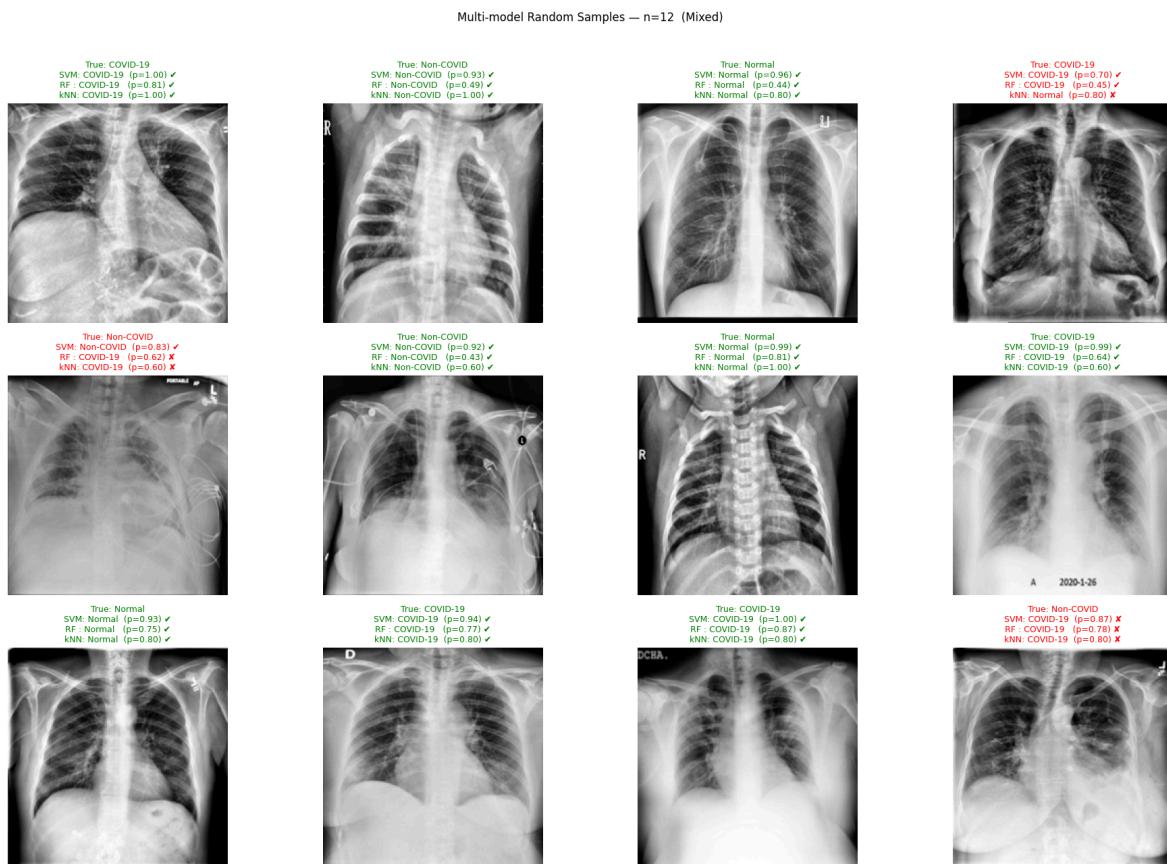
## 4.5 Analisis Lintas Model

Analisis **overlap kesalahan** (Gbr. 4.10) menunjukkan 87 sampel salah bersama-sama oleh ketiga model ( $SVM \cap RF \cap kNN$ ). Mayoritas berasal dari kelas Non-COVID, menegaskan bahwa representasi fitur yang digunakan masih belum cukup membedakan antara pneumonia non-COVID dan COVID-19.



**Gbr. 4.10 Overlap Kesalahan Lintas Model (Test)**

Visualisasi gabungan acak lintas model (Gbr. 4.11) memperlihatkan setiap citra disertai prediksi SVM, RF, dan kNN serta confidence-nya. Warna merah pada judul menandai ada minimal satu model yang salah. Dari observasi, SVM lebih konsisten sementara RF dan kNN lebih fluktuatif terhadap variasi tekstur dan noise.



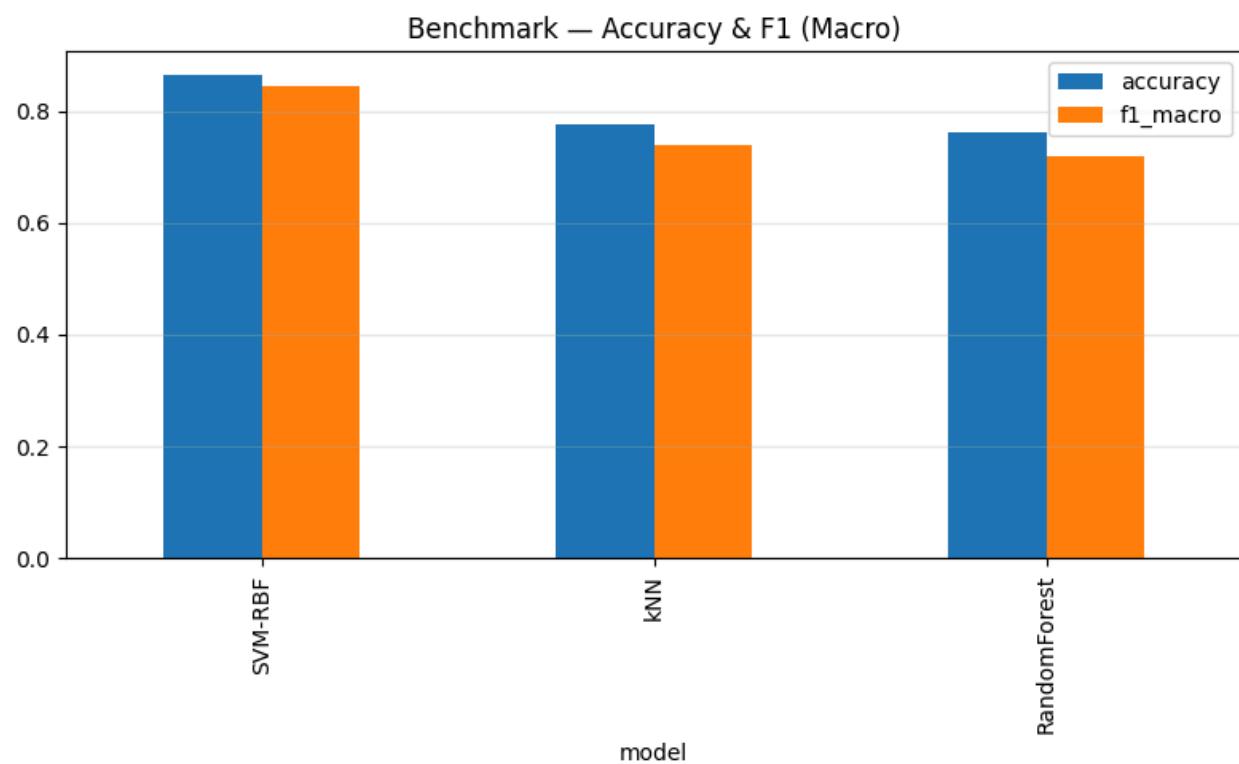
**Gbr. 4.11 Grid Multi-Model Random Samples (Benar vs Salah)**

#### 4.6 Benchmark Performa Model

Bagian benchmark (Tabel 4.1 dan Gbr. 4.12) menampilkan perbandingan metrik utama dan efisiensi:

Tabel 4.1. Benchmark — Accuracy dan F1 (Macro)

Model	Accuracy	F1 (macro)	ROC-AUC	Latency (ms/sampel)	Ukuran Model (MB)
SVM-RBF	0.8627	0.8432	0.9601	14.96	95.7
RandomForest	0.7609	0.7188	0.9283	0.14	21.16
kNN	0.7757	0.7389	0.8828	127.47	127.47



Gbr. 4.12 Benchmark — Accuracy & F1 (Macro)

SVM menjadi model terbaik secara keseluruhan, menyeimbangkan akurasi tinggi dan stabilitas probabilitas. Random Forest lebih ringan dan cepat, cocok untuk deployment real-time, sedangkan kNN memberikan intuisi lokal namun boros memori dan waktu inferensi.

## Tahap 5: Eksperimen Deep Learning (Custom CNN + LoRA)

### 5.1 Latar Belakang dan Arsitektur Model

Setelah mengeksplorasi pendekatan *machine learning* klasik, tahap selanjutnya adalah menerapkan *Deep Learning* menggunakan *Convolutional Neural Network* (CNN). Berbeda dengan pendekatan sebelumnya yang memisahkan ekstraksi fitur (HOG) dan klasifikasi, CNN melakukan pembelajaran fitur secara *end-to-end*.

Pada eksperimen ini, kami membangun arsitektur Custom CNN dari nol (*from scratch*) yang dipadukan dengan teknik Low-Rank Adaptation (LoRA) pada lapisan *fully-connected*. Tujuannya adalah menciptakan model yang ringan (parameter efisien) namun tetap mampu menangkap pola spasial pada citra X-ray.

Arsitektur yang dibangun terdiri dari 4 blok konvolusi. Setiap blok memiliki urutan layer: Conv2D -> BatchNormalization -> ReLU -> MaxPooling. Total parameter yang dapat dilatih (*trainable parameters*) berhasil ditekan menjadi sekitar 405.731 parameter, menjadikannya model yang sangat ringan (hanya berukuran fisik sekitar 1.8 MB).

### 5.2 Implementasi LoRA (Low-Rank Adaptation)

Inovasi utama pada tahap ini adalah penggantian lapisan *Dense* standar dengan lapisan LoRADense. Alih-alih melatih matriks bobot penuh yang berdimensi besar, kami membekukan bobot dasar (*frozen weights*) dan menyuntikkan dua matriks berperingkat rendah (*low-rank matrices*) yang dapat dilatih

Teknik ini secara signifikan mengurangi beban komputasi saat *backpropagation* dan mencegah perubahan drastis pada bobot model, yang diharapkan dapat meningkatkan stabilitas generalisasi pada dataset yang terbatas.

### 5.3 Konfigurasi Pelatihan

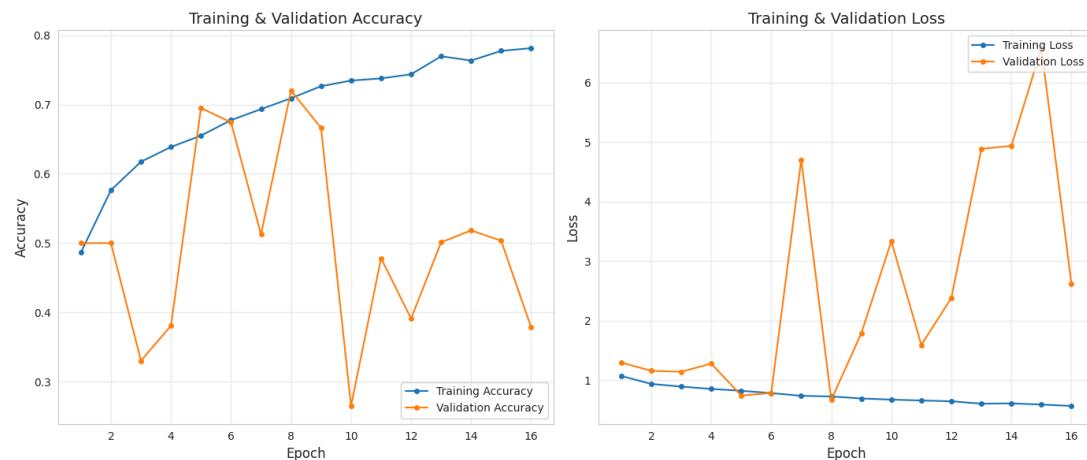
Model dilatih menggunakan data yang sama dengan tahap sebelumnya (Stratified Split). Mengingat ketidakseimbangan kelas (COVID-19 mendominasi 50% data), kami menerapkan Class Weights secara ketat pada fungsi *loss*:

- Bobot COVID-19: ~0.67 (ditekan karena data banyak)
- Bobot Non-COVID & Normal: ~1.33 (dinaikkan agar model lebih memperhatikan kelas minoritas)

Strategi *training* menggunakan *optimizer* Adam dengan *learning rate* awal 0.001, dilengkapi dengan *Early Stopping* (patience 8 epoch) untuk mencegah *overfitting* yang parah, serta *ReduceLROnPlateau* jika validasi stagnan.

#### 5.4 Evaluasi Kinerja Model

Berdasarkan hasil pelatihan, model dihentikan pada epoch ke-16 (mengembalikan bobot terbaik dari epoch 8) karena *validation loss* mulai tidak stabil.



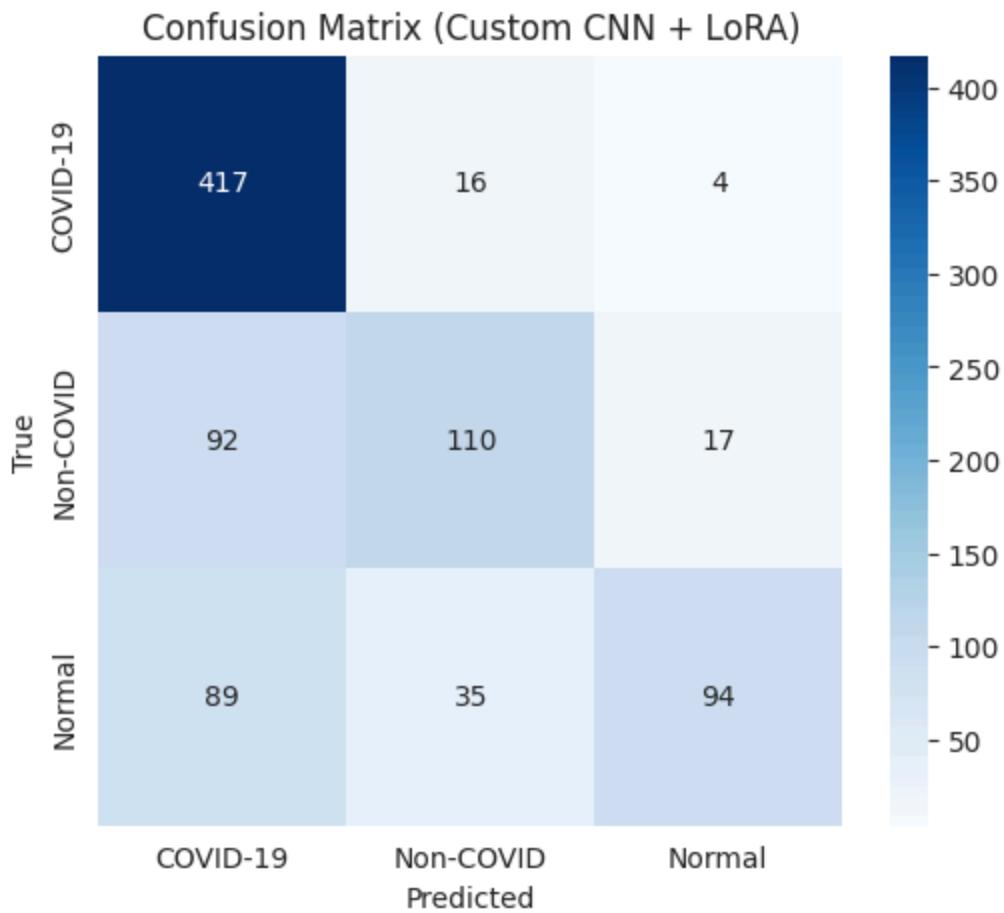
Gbr 5.1. Grafik Akurasi dan Loss selama Pelatihan (Train vs Validation).

Catatan Analisis: Grafik di atas menunjukkan fenomena *overfitting* yang cukup jelas setelah epoch ke-8. Garis validation loss (orange) melonjak tajam menjauhi training loss (biru), mengindikasikan model mulai menghafal data latih namun gagal menggeneralisasi pola pada data validasi. Mekanisme Early Stopping berhasil mencegah degradasi lebih lanjut.

Pada pengujian akhir menggunakan *Test Set* (874 citra), model mencatat akurasi global sebesar **71.05%**. Berikut adalah detail performa per kelas:

Tabel 5.1. Laporan Klasifikasi Custom CNN + LoRA

Kelas	Precision	Recall	F1-Score
<b>COVID-19</b>	0.6973	<b>0.9542</b>	0.8058
<b>Non-COVID</b>	0.6832	0.5023	0.5789
<b>Normal</b>	0.8174	0.4312	0.5646
<b>Akurasi Total</b>			<b>71.05%</b>



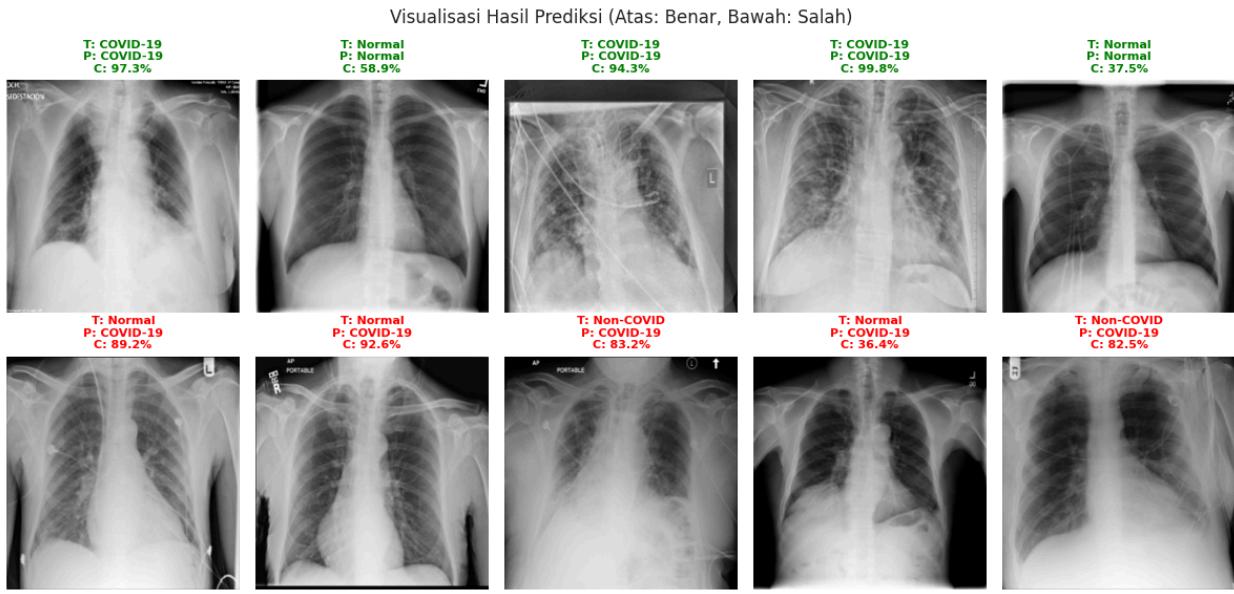
**Gbr 5.2. Confusion Matrix pada Data Uji.**

Analisis Confusion Matrix menunjukkan karakteristik menarik:

1. **Sensitivitas Tinggi terhadap COVID-19:** Model berhasil mengenali 95% kasus positif COVID-19 (Recall 0.95). Ini sangat baik untuk keperluan skrining medis agar tidak ada kasus positif yang terlewat.
2. **Kebingungan pada Kelas Normal/Non-COVID:** Model mengalami kesulitan membedakan kelas Normal dan Non-COVID, dengan banyak sampel dari kedua kelas ini yang salah diprediksi sebagai COVID-19 (Tingkat *False Positive* cukup tinggi). Hal ini kemungkinan disebabkan oleh arsitektur CNN yang terlalu sederhana (shallow) sehingga belum mampu menangkap fitur *fine-grained* yang membedakan paru-paru normal dengan pneumonia non-COVID.

## 5.5 Visualisasi Prediksi Kualitatif

Untuk memvalidasi hasil metrik, kami melakukan inspeksi visual terhadap sampel prediksi.



**Gbr 5.3. Sampel Prediksi (Baris Atas: Prediksi Benar, Baris Bawah: Prediksi Salah).**

Pada baris bawah (kasus salah), terlihat bahwa model cenderung "terlalu waspada" dan memprediksi citra sebagai COVID-19 meskipun label aslinya adalah Normal atau Non-COVID. Ini konsisten dengan temuan *Recall* COVID-19 yang sangat tinggi namun *Precision* yang moderat.

Kesimpulan Tahap 5:

Penerapan Custom CNN dengan LoRA menghasilkan model yang sangat efisien (ringan) dengan kemampuan deteksi COVID-19 yang sensitif (*Recall* 95%). Namun, secara keseluruhan akurasinya (71%) masih di bawah pendekatan SVM-RBF (86%) pada Tahap 3. Hal ini menunjukkan bahwa untuk dataset dengan ukuran ini, fitur terstruktur (HOG) masih lebih robust dibandingkan fitur yang dipelajari dari nol oleh CNN kecil, atau diperlukan arsitektur CNN yang lebih dalam (transfer learning) untuk meningkatkan performa.

## Tahap 6: Eksperimen Deep Learning Lanjutan

### 6.1 Custom CNN + LoRA dengan Augmentasi Data

Konfigurasi pertama menggunakan arsitektur Custom CNN yang dirancang khusus untuk dataset COVID-QU-Ex. Arsitektur ini kemudian dipadukan dengan pendekatan Low-Rank Adaptation (LoRA) sehingga sebagian besar bobot konvolusional dapat dibekukan, dan adaptasi ke domain citra X-ray dilakukan melalui penambahan sejumlah kecil parameter ber-rank rendah.

### a. Arsitektur Custom CNN

Secara garis besar, arsitektur Custom CNN yang digunakan terdiri dari:

- Beberapa blok berurutan Conv2D → Batch Normalization → ReLU → MaxPooling, dengan jumlah filter yang meningkat secara bertahap (misalnya 32 → 64 → 128 → 256).
- Sebuah lapisan Global Average Pooling untuk mereduksi peta fitur spasial menjadi satu vektor fitur global.
- Satu atau dua lapisan Dense dengan dropout sebagai regularisasi.
- Lapisan output Dense + Softmax dengan 3 neuron yang merepresentasikan tiga kelas: COVID-19, Non-COVID, dan Normal.

Dengan struktur seperti ini, Custom CNN diharapkan mampu menangkap pola lokal maupun global pada citra chest X-ray, tetapi tetap memiliki jumlah parameter yang relatif moderat.

### b. Penerapan LoRA pada Custom CNN

Pada arsitektur Custom CNN ini, LoRA diterapkan terutama pada lapisan-lapisan konvolusional di bagian menengah hingga akhir (misalnya pada blok dengan 128 dan 256 filter). Secara konsep:

- Bobot konvolusi dasar ( $W_0$ ) dibekukan dan tidak di-update selama training.
- Ditambahkan dua matriks ber-rank rendah ( $A$ ) dan ( $B$ ) dengan rank kecil (misalnya 4–8), sehingga bobot efektif menjadi:  
[  
$$W = W_0 + A \cdot B$$
  
]
- Selama proses training, hanya parameter ( $A$ ) dan ( $B$ ) (serta beberapa lapisan atas seperti classifier) yang di-update.

Dengan skema ini, proses pelatihan menjadi lebih **efisien** dan **stabil**, karena mayoritas parameter besar tetap mempertahankan struktur awal, sementara penyesuaian terhadap karakteristik dataset dilakukan oleh adaptor LoRA yang jauh lebih kecil.

### c. Augmentasi Data

Berbeda dengan varian Custom CNN + LoRA tanpa augmentasi (yang pada analisis akhir justru terbukti menjadi deep model terbaik di antara konfigurasi CNN), bagian ini secara khusus menggunakan varian dengan augmentasi data pada set latih. Tujuannya adalah:

- menambah variasi citra yang dilihat model,
- mengurangi risiko overfitting terhadap pola spesifik pada data train.

Beberapa bentuk augmentasi yang digunakan antara lain:

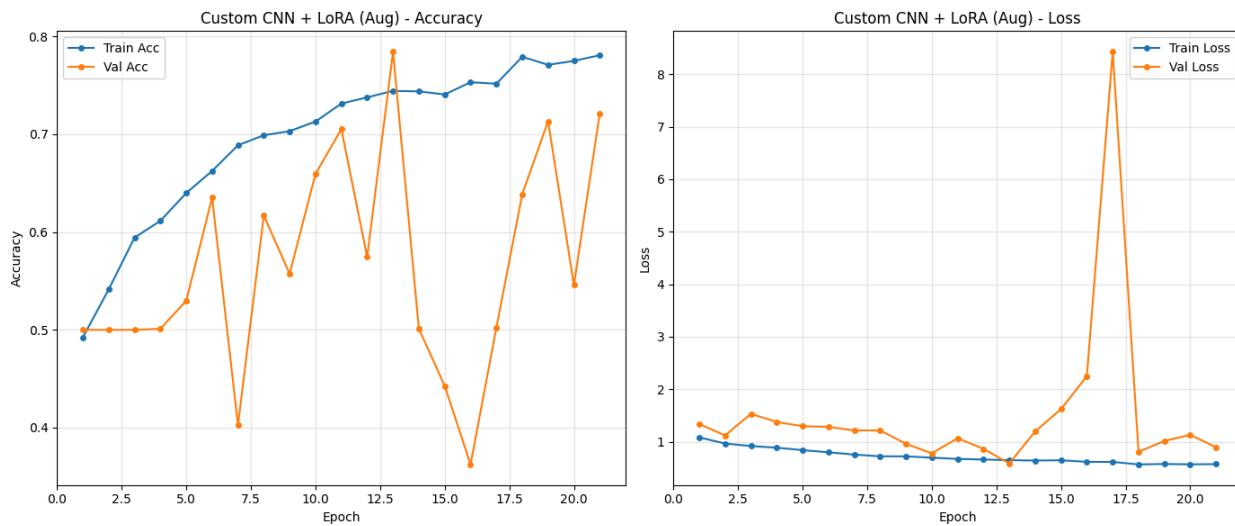
- Random horizontal flip dengan probabilitas tertentu, untuk mensimulasikan variasi orientasi.
- Rotasi kecil (misalnya  $\pm 10^\circ$ ) untuk mengakomodasi perubahan posisi pasien.
- Random zoom dalam rentang terbatas (zoom in/out ringan).
- (Opsional) Sedikit variasi brightness/contrast yang tetap dijaga agar tidak mengubah makna klinis citra.

Seluruh augmentasi ini hanya diterapkan pada data train. Sementara itu, data validasi dan test tetap diproses secara konsisten tanpa augmentasi, untuk menjaga kejujuran evaluasi model.

#### d. Proses Training

Proses pelatihan Custom CNN + LoRA dengan augmentasi secara umum meliputi:

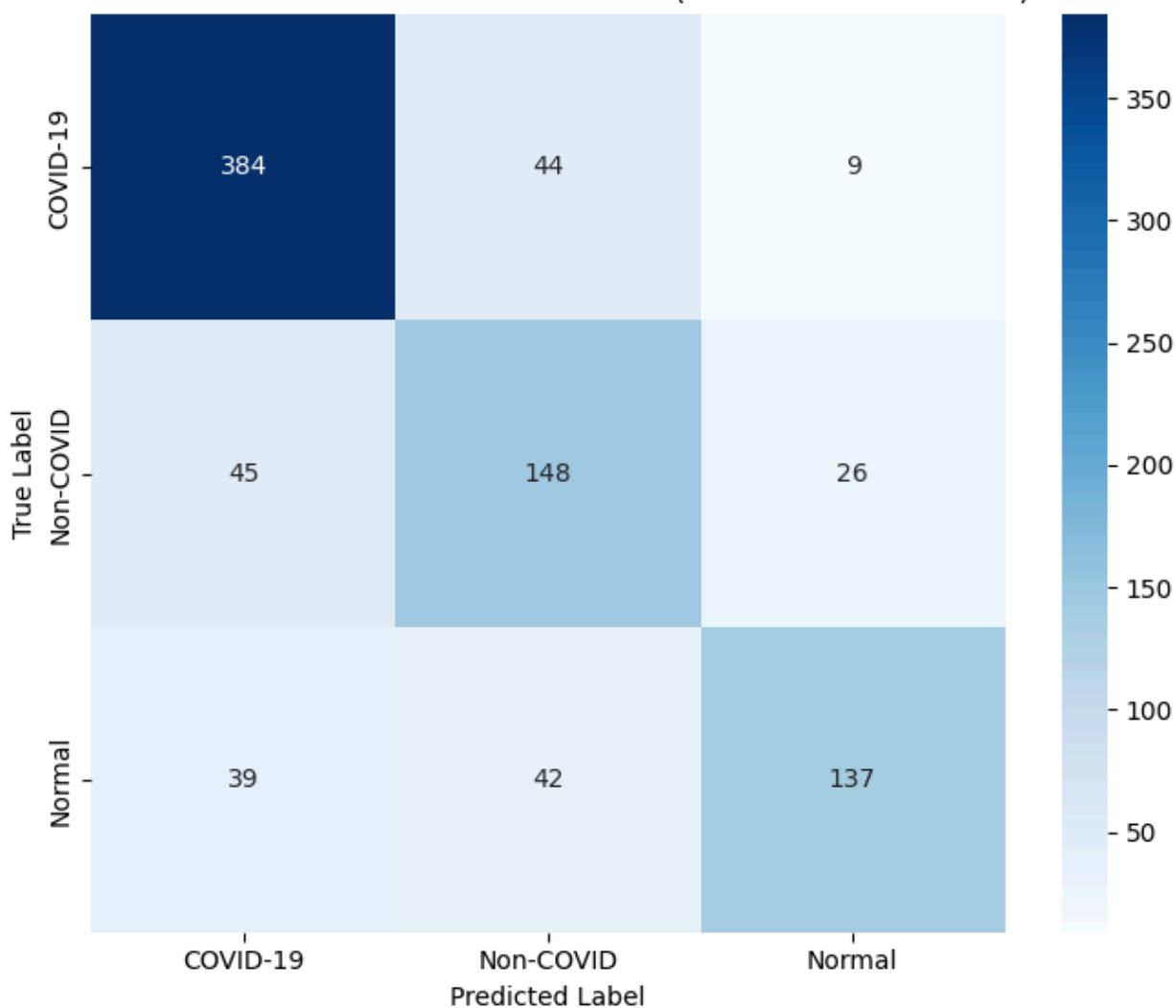
- Optimizer: Adam atau AdamW dengan learning rate awal kecil (misalnya  $10^{-4}$ ).
- Loss function: categorical cross-entropy yang dikombinasikan dengan class weight sesuai distribusi kelas.
- Batch size: sekitar 16–32, menyesuaikan kapasitas GPU/komputasi.
- Callback:
  - EarlyStopping berbasis validation loss,
  - ModelCheckpoint untuk menyimpan bobot terbaik selama training.



**Gbr. 6.1 Kurva akurasi dan loss pelatihan model Custom CNN + LoRA dengan augmentasi data**

Secara hasil, konfigurasi ini memberikan performa yang cukup baik dan menunjukkan bahwa Custom CNN + LoRA mampu belajar representasi yang relevan. Namun, ketika dibandingkan secara komprehensif, varian Custom CNN + LoRA tanpa augmentasi justru sedikit lebih unggul pada test set. Meskipun demikian, varian dengan augmentasi ini tetap penting sebagai pembanding karena mendekati praktik umum pelatihan CNN pada citra medis.

Confusion Matrix - Custom CNN + LoRA (WITH AUGMENTATION)



**Gbr. 6.1 Confusion matrix model Custom CNN + LoRA dengan augmentasi pada test set.**

Meskipun demikian, varian dengan augmentasi ini tetap penting sebagai pembanding karena mendekati praktik umum pelatihan CNN pada citra medis.

## 6.2 Transfer Learning dengan DenseNet121 dan LoRA

Pada tahap-tahap sebelumnya, model berbasis fitur klasik (HOG + SVM RBF) terbukti mampu memberikan performa yang sangat baik, sementara beberapa varian CNN yang dilatih dari awal cenderung lebih sensitif terhadap ukuran data dan relatif mudah mengalami overfitting. Di sisi lain, pendekatan deep learning tetap menarik karena mampu belajar langsung dari citra tanpa memerlukan rekayasa fitur manual yang rumit.

Untuk menjembatani kedua hal tersebut, pada tahap ini digunakan arsitektur DenseNet121 yang sudah di-pretrain pada ImageNet, kemudian disesuaikan dengan dataset COVID-QU-Ex

menggunakan pendekatan Low-Rank Adaptation (LoRA) pada bagian kepala klasifikasi. Secara sederhana, DenseNet dimanfaatkan sebagai “penyedia fitur umum” yang sudah belajar berbagai pola visual dari jutaan gambar, sementara LoRA bertugas melakukan penyesuaian ringan agar model lebih cocok dengan citra X-ray dada.

Pendekatan ini dipilih karena:

- tidak perlu melatih seluruh parameter DenseNet dari awal,
- jumlah parameter yang benar-benar di-update jauh lebih sedikit,
- risiko overfitting lebih kecil,
- proses pelatihan lebih ringan secara komputasi.

Harapannya, kita tetap memperoleh keunggulan deep learning (belajar fitur langsung dari data) tetapi dengan stabilitas yang lebih baik pada data dengan ukuran terbatas.

Agar hasil antar-model mudah dibandingkan, alur pra-pemrosesan citra dibuat konsisten dengan tahap-tahap sebelumnya. Setiap citra chest X-ray terlebih dahulu:

1. dikonversi ke grayscale,
2. di-resize ke ukuran  $224 \times 224$  piksel,
3. ditingkatkan kontrasnya menggunakan CLAHE,
4. dipotong secara heuristik (heuristic lung crop) agar fokus ke area paru,
5. dinormalisasi ke rentang nilai  $[0, 1]$ .

Pembagian data ke train, validation, dan test dilakukan secara stratified dengan seed tetap, sehingga seluruh model (baik klasik maupun deep learning) dievaluasi pada pembagian data yang persis sama.

Karena DenseNet121 pre-trained pada citra RGB, citra grayscale diubah menjadi 3 kanal ( $H \times W \times 3$ ) dengan cara menduplikasi kanal abu-abu. Selanjutnya, digunakan augmentasi sederhana (misalnya flip horizontal dan sedikit rotasi/zoom) untuk menambah variasi pola tanpa mengubah makna klinis citra.

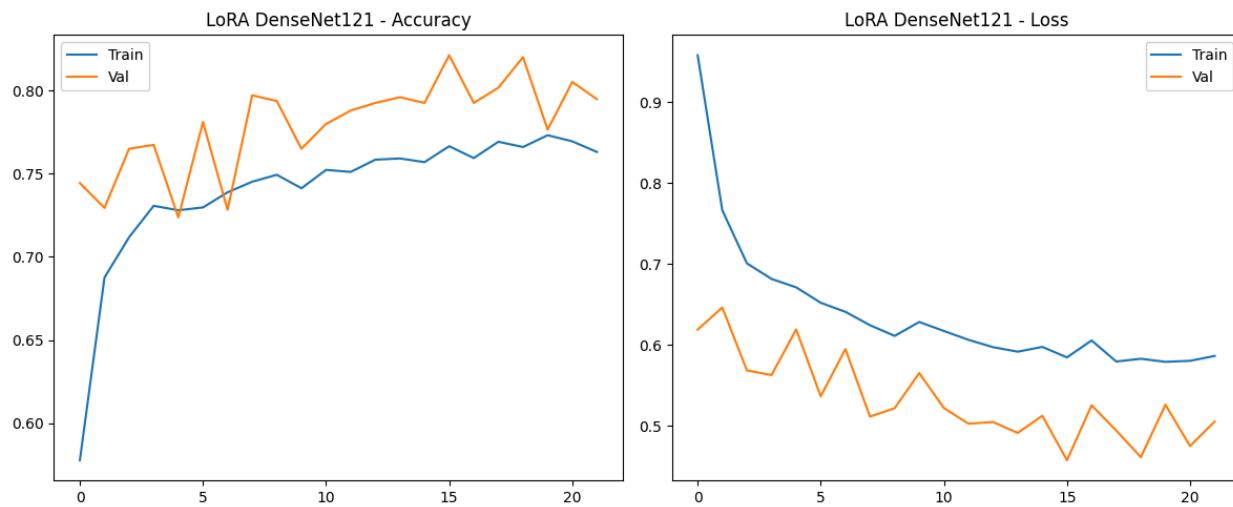
Bagian utama model terdiri atas:

- Backbone DenseNet121 dengan `include_top=False` dan bobot pre-trained ImageNet. Pada tahap ini backbone dibekukan (`trainable=False`), sehingga berfungsi sebagai ekstraktor fitur tetap.
- GlobalAveragePooling2D, untuk merangkum fitur spasial menjadi vektor tunggal.
- Dropout, untuk membantu mengurangi overfitting.
- Lapisan kepala (head) dengan LoRA, yaitu lapisan dense yang tidak melatih keseluruhan matriks bobot, tetapi hanya menambahkan dua matriks kecil ber-rank rendah sebagai adaptor. Matriks utama hasil pre-training tetap dibekukan.

Terakhir digunakan lapisan Dense dengan softmax untuk memetakan fitur menjadi tiga kelas: COVID-19, Non-COVID, dan Normal.

Model dilatih menggunakan optimizer Adam dengan learning rate kecil. Class weight dihitung berdasarkan distribusi kelas untuk mengurangi bias terhadap kelas mayoritas. Beberapa callback yang digunakan meliputi:

- **EarlyStopping** untuk menghentikan training ketika validation loss tidak membaik lagi,
- **ModelCheckpoint** untuk menyimpan bobot terbaik,
- **ReduceLROnPlateau** untuk menurunkan learning rate ketika model mulai “stagnan”.

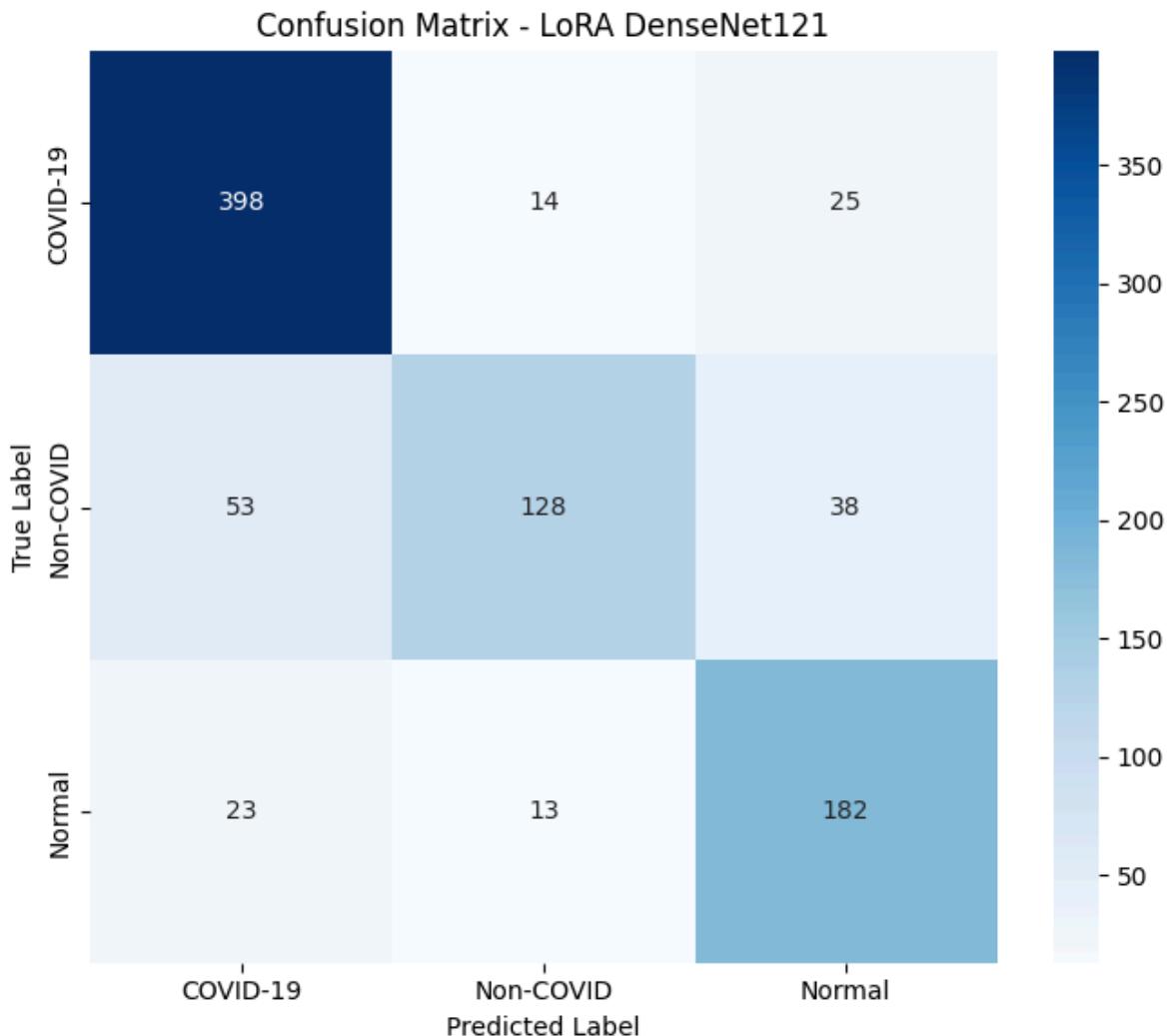


**Gbr. 6.2 Kurva akurasi dan loss pelatihan model DenseNet121 + LoRA pada data train dan validation.**

Hasil pelatihan menunjukkan kurva yang relatif stabil. Akurasi train dan validation meningkat bersama-sama kemudian cenderung mendatar pada nilai yang cukup tinggi. Validation loss turun lalu mencapai plateau tanpa jarak yang terlalu besar dengan training loss, sehingga tidak tampak tanda overfitting berat.

Evaluasi pada test set menunjukkan bahwa DenseNet121 + LoRA:

- mencapai akurasi sekitar 81%,
- menghasilkan macro-F1 di kisaran 0,78.



**Gbr. 6.2 Confusion matrix model DenseNet121 + LoRA pada test set.**

Dari classification report dan confusion matrix, pola yang muncul adalah:

- **Kelas COVID-19**  
Precision dan recall cukup tinggi. Sebagian besar citra COVID-19 dikenali dengan benar, dan jumlah kasus COVID-19 yang terlewat (false negative) relatif kecil. Dari sisi klinis, ini penting karena melewatkannya kasus positif berpotensi berbahaya.
- **Kelas Normal**  
Recall kelas Normal juga baik, menandakan banyak citra paru sehat yang berhasil dikenali sebagai Normal.
- **Kelas Non-COVID**  
Kelas ini tetap menjadi yang paling sulit. Citra Non-COVID kadang diprediksi sebagai COVID-19 atau Normal, sehingga F1-score kelas Non-COVID lebih rendah dibanding dua kelas lain.

Beberapa contoh citra yang salah klasifikasi memberikan gambaran yang cukup intuitif:

1. Citra Non-COVID dengan infiltrat difus di kedua paru terkadang diprediksi sebagai COVID-19. Secara radiologis, pola seperti ini memang mirip pneumonia COVID-19.
2. Citra Non-COVID dengan kelainan sangat ringan kadang masuk ke kelas Normal, menunjukkan model masih kesulitan membedakan paru yang benar-benar normal dengan kelainan yang halus.
3. Artefak seperti kabel, tulisan, atau posisi pasien yang kurang ideal dapat mengganggu pola paru sehingga prediksi menjadi tidak stabil.

Selain itu, ada beberapa kasus salah prediksi dengan **confidence yang cukup tinggi**. Ini perlu diperhatikan bila model akan digunakan sebagai sistem pendukung keputusan, karena prediksi yang salah tetapi sangat yakin dapat berpotensi menyesatkan jika tidak dilengkapi dengan penilaian klinis.

Secara singkat, **model ini sudah cukup baik untuk kelas COVID-19 dan Normal**, sementara kesalahan utama masih terkonsentrasi pada kelas Non-COVID, khususnya pada kasus yang secara visual memang ambigu.

Dari hasil di atas, dapat disimpulkan bahwa **DenseNet121 + LoRA** merupakan salah satu kombinasi deep learning yang kuat untuk dataset ini. Dibandingkan CNN “sederhana” yang dilatih dari nol, model ini:

- lebih stabil,
- lebih cepat dilatih,
- dan memberikan performa yang konsisten tinggi.

Namun, setelah seluruh eksperimen deep learning selesai, posisi DenseNet121 + LoRA ternyata **bukan lagi model deep learning terbaik**. Variasi **Custom CNN + LoRA tanpa augmentasi** (lihat ringkasan di Tahap 8) mampu mencapai akurasi test sekitar **84,44%** dengan macro-F1 yang sedikit lebih tinggi. Dengan kata lain, DenseNet121 + LoRA menjadi **salah satu kandidat kuat**, tetapi masih sedikit di bawah CNN+LoRA terbaik.

Dibandingkan **HOG + SVM RBF**, performanya berada sedikit di belakang. Pendekatan berbasis fitur klasik ini tetap unggul dari sisi angka murni, tetapi DenseNet121 + LoRA memiliki kelebihan berupa **pipeline end-to-end** yang lebih mudah diintegrasikan dengan berbagai teknik interpretabilitas (misalnya Grad-CAM) dan sistem deep learning lain.

Keterbatasan utama DenseNet121 + LoRA tetap berada pada pembedaan kelas Non-COVID, yang secara radiologis memang berada di “tengah” antara COVID-19 dan Normal. Dalam praktik klinis, model ini lebih tepat diposisikan sebagai **alat bantu triase**: menandai kasus yang sangat mungkin COVID-19 atau sangat jelas Normal, sementara kasus-kasus ambigu tetap perlu ditinjau langsung oleh radiolog.

### 6.3 Vision Transformer (ViT)

Setelah mencoba pendekatan CNN klasik serta transfer learning dengan DenseNet, pada tahap ini dilakukan serangkaian eksperimen dengan **Vision Transformer (ViT)**. Berbeda dengan CNN yang mengandalkan operasi konvolusi, ViT memecah gambar menjadi patch kecil dan memprosesnya sebagai urutan (sequence), mirip cara model transformer di NLP memproses token teks.

Dalam penelitian ini, dilakukan **dua jenis eksperimen ViT**:

1. **ViT Keras murni**, yaitu model ViT yang dibangun dengan Keras dan **dilatih dari awal (tanpa pre-training besar)** pada dataset COVID-QU-Ex.
2. **ViT pre-trained (HuggingFace)**, yaitu model **google/vit-base-patch16-224-in21k** yang sudah di-pretrain pada dataset skala besar, kemudian di-fine-tune pada dataset yang sama (COVID-QU-Ex).

Tujuan utamanya adalah:

1. Melihat sejauh mana ViT yang dilatih dari nol mampu belajar dari dataset chest X-ray berskala menengah.
2. Mengukur seberapa besar peningkatan ketika **pre-training besar** dimanfaatkan (melalui model ViT pre-trained dari HuggingFace).
3. Membandingkan karakteristik error ViT (baik murni maupun pre-trained) dengan model CNN dan model klasik seperti SVM.

Dengan demikian, tahap ini bersifat cukup eksploratif dan berfungsi untuk memahami batas dan potensi ViT dalam skenario data yang realistik.

Pra-pemrosesan citra pada ViT Keras dibuat konsisten dengan tahap-tahap sebelumnya: citra dikonversi ke grayscale, di-resize ke  $224 \times 224$ , diberi CLAHE, dilakukan lung crop heuristik, dan dinormalisasi ke  $[0, 1]$ . Setelah itu, citra dipecah menjadi patch berukuran  $16 \times 16$  piksel. Setiap patch kemudian:

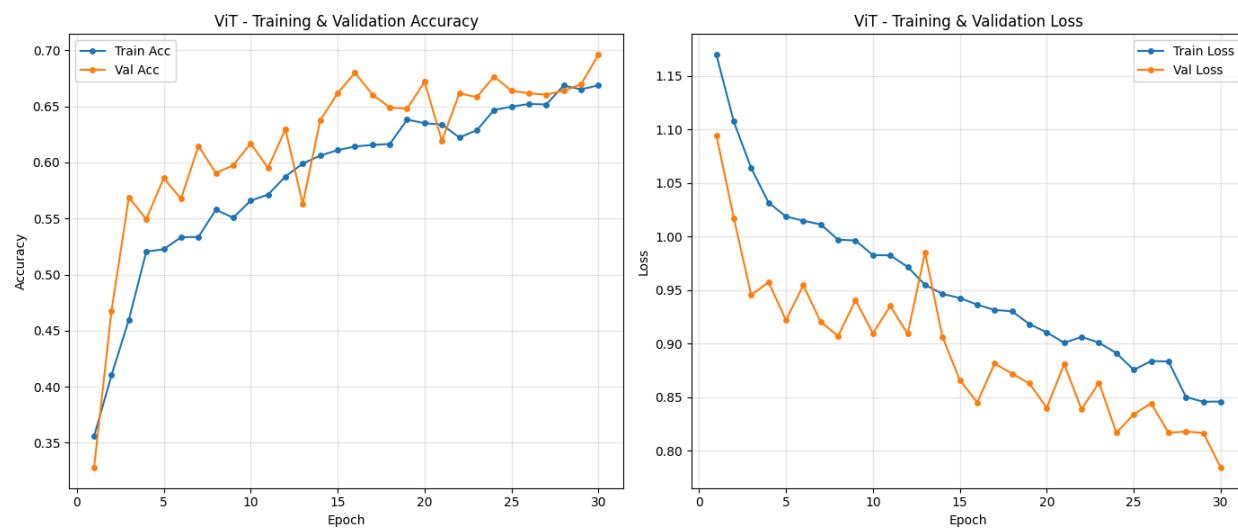
- di-*flatten* menjadi vektor,
- diproyeksikan ke ruang embedding berdimensi tertentu melalui lapisan Dense,
- ditambahkan **positional embedding** untuk menyimpan informasi posisi patch di dalam gambar.

Urutan embedding patch ini kemudian masuk ke beberapa blok **Transformer Encoder**. Setiap blok encoder terdiri atas:

- layer normalization,
- multi-head self-attention,
- MLP kecil dengan aktivasi non-linear dan dropout.

Self-attention memungkinkan model untuk “memperhatikan” hubungan antar patch yang letaknya berjauhan, misalnya antara kedua paru atau antara area lesi dan area sekitarnya. Setelah melewati tumpukan encoder, representasi patch diringkas (misalnya dengan average pooling) menjadi satu vektor yang kemudian diproses melalui MLP head untuk menghasilkan prediksi tiga kelas.

Hyperparameter ViT Keras dibuat **moderat**, menyesuaikan ukuran dataset dan kapasitas GPU di lingkungan eksperimen. Optimizer yang digunakan adalah **AdamW** dengan weight decay, dropout relatif tinggi, dan **class weight** untuk mengatasi ketidakseimbangan kelas. EarlyStopping dan ModelCheckpoint kembali digunakan untuk menghentikan pelatihan ketika tidak ada perbaikan berarti di validation loss.

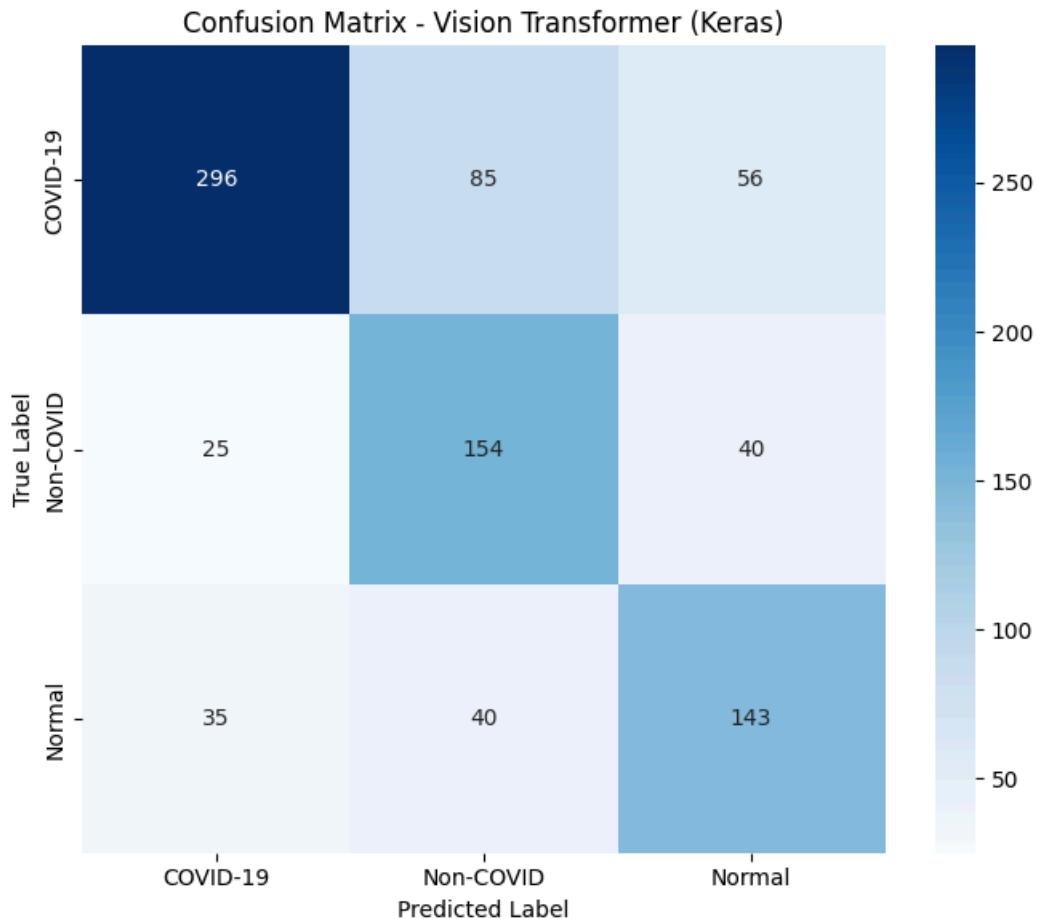


**Gbr. 6.3 Kurva akurasi dan loss pelatihan model Vision Transformer (ViT) Keras yang dilatih dari awal tanpa pre-training besar.**

Secara umum, ViT Keras **mampu belajar** dari data, tetapi performanya belum dapat mengimbangi model-model terbaik lain. Akurasi test berada di kisaran sekitar **dua pertiga** dari sampel yang diklasifikasikan dengan benar, dan **macro-F1** masih berada di bawah DenseNet+LoRA maupun HOG+SVM.

Polanya:

- Akurasi train dan validation tidak terlalu berjauhan,
- Tidak tampak overfitting parah,
- Namun justru cenderung **sedikit underfitting**, menandakan kapasitas model belum sepenuhnya terpakai, kemungkinan karena data relatif terbatas untuk arsitektur transformer yang besar.



**Gbr. 6.3 Confusion matrix model Vision Transformer (ViT) Keras yang dilatih dari awal pada test set.**

Dari confusion matrix, error ViT Keras cenderung **tersebar cukup merata di semua kelas**. Tidak ada kelas yang benar-benar unggul, tetapi juga tidak ada yang jatuh sangat buruk. Meskipun demikian, secara keseluruhan semua metrik masih berada di bawah model terbaik seperti Custom CNN + LoRA (tanpa augmentasi) dan DenseNet+LoRA.

Beberapa temuan penting:

1. **Kelas Normal**

Cukup banyak citra Normal yang diprediksi sebagai COVID-19 atau Non-COVID. Hal ini menunjukkan bahwa ViT dari nol membutuhkan data lebih besar untuk benar-benar memahami variasi paru normal yang “halus”.

2. **COVID-19 vs Non-COVID**

Kebingungan antara kedua kelas patologis ini kembali muncul. Pola lesi yang difus dan bilateral sulit dibedakan hanya dari intensitas dan distribusi opasitas, sehingga prediksi kadang bergeser ke kelas yang salah.

### 3. Ukuran patch

Dengan patch relatif besar ( $16 \times 16$ ), detail kecil seperti ground-glass opacity berukuran kecil dapat “tenggelam” dalam satu patch bersama bagian paru yang normal, sehingga sinyal kelainan menjadi lemah.

### 4. Confidence yang belum terkalibrasi

Beberapa prediksi salah masih memiliki probabilitas tinggi. Karena model underfitting, distribusi probabilitasnya belum terkalibrasi dengan baik, sehingga angka confidence belum bisa dijadikan indikator kepercayaan model secara langsung.

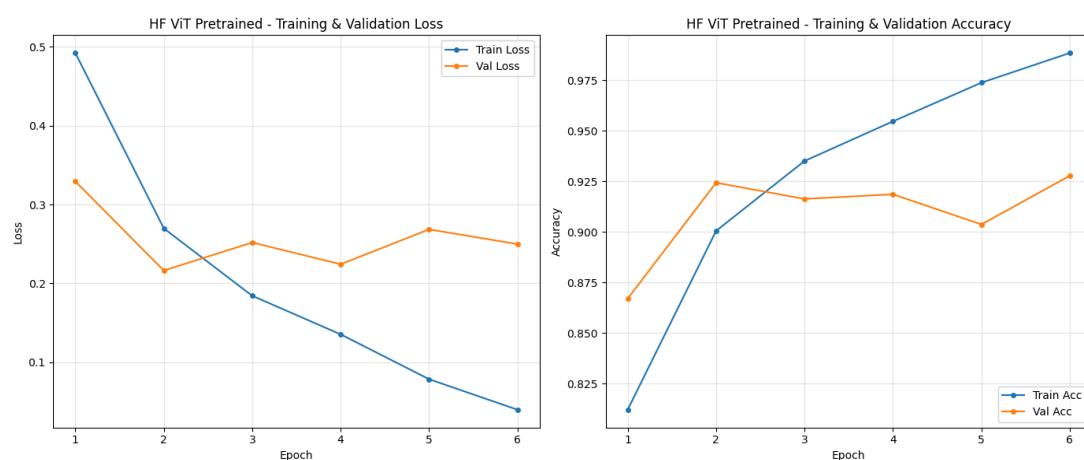
Secara keseluruhan, keterbatasan utama ViT Keras di sini adalah **ketiadaan pre-training besar** dan ukuran dataset yang relatif terbatas untuk arsitektur transformer.

## 6.4 Vision Transformer (ViT) pre-trained (HuggingFace)

Untuk melihat efek pre-training secara lebih eksplisit, dilakukan eksperimen tambahan dengan **model ViT pre-trained dari HuggingFace**, yaitu:

- **Arsitektur:** google/vit-base-patch16-224-in21k
- Framework: PyTorch (transformers + timm)
- Strategi: **fine-tuning** pada dataset COVID-QU-Ex dengan:
  - preprocessing yang sama (CLAHE + lung crop),
  - citra grayscale diubah menjadi 3 kanal,
  - augmentasi sederhana (flip horizontal) hanya pada train set,
  - class weight untuk mengatasi ketidakseimbangan kelas.

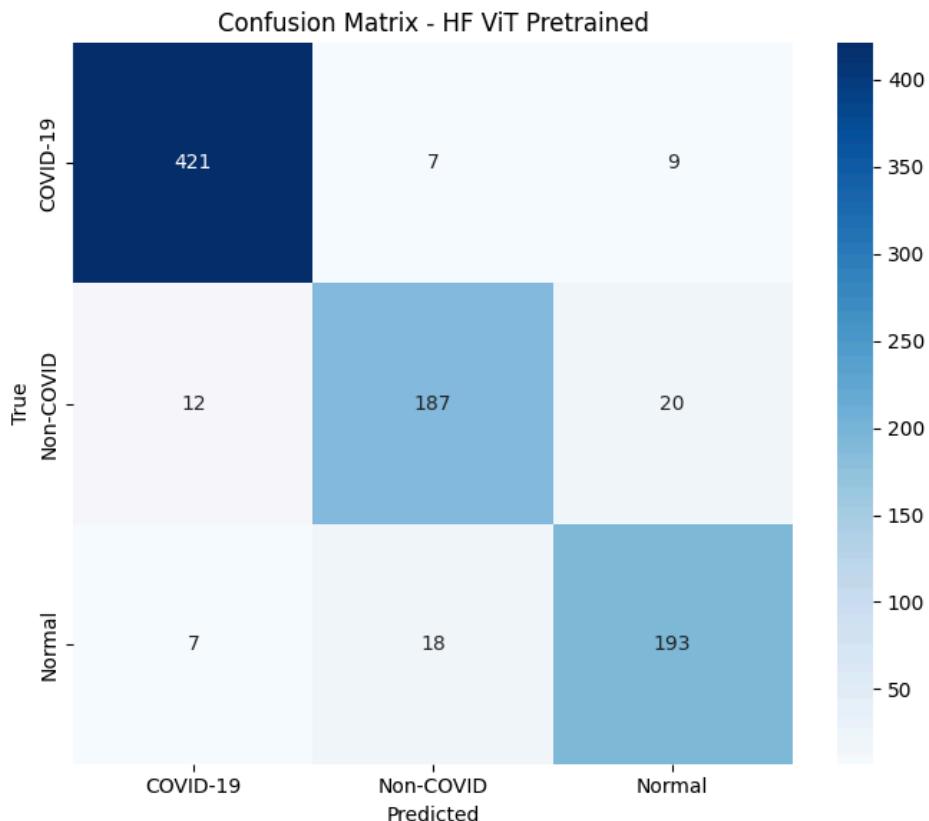
Head klasifikasi diadaptasi menjadi 3 kelas (COVID-19, Non-COVID, Normal), dan seluruh model di-fine-tune dengan learning rate kecil menggunakan optimizer AdamW serta scheduler ReduceLROnPlateau dan early stopping.



Gbr. 6.4 Kurva akurasi dan loss pelatihan model HF ViT pre-trained (google/vit-base-patch16-224-in21k) setelah fine-tuning pada dataset COVID-QU-Ex.

Secara kualitatif, hasilnya menunjukkan bahwa:

- **Performa ViT pre-trained secara konsisten lebih baik daripada ViT Keras yang dilatih dari nol**, baik dari sisi akurasi maupun macro-F1.
- Meskipun demikian, ViT pre-trained belum mampu melampaui kombinasi **Custom CNN + LoRA (tanpa augmentasi)** yang menjadi deep model terbaik pada eksperimen ini, dan umumnya berada di kisaran yang mirip atau sedikit di bawah DenseNet121 + LoRA pada beberapa metrik.



**Gbr. 6.4 Confusion matrix model Vision Transformer (ViT) Keras yang dilatih dari awal pada test set.**

- Pola error masih menunjukkan bahwa **kelas Non-COVID tetap paling sulit**, tetapi jumlah error kasar berkurang dibanding ViT murni.

Eksperimen ini menguatkan dugaan bahwa pre-training berskala besar sangat penting untuk membuat ViT kompetitif pada dataset citra medis yang ukuran dan variasinya terbatas.

## 6.5 Catatan dan peluang pengembangan

Meskipun ViT (baik murni maupun pre-trained) belum menjadi model terbaik di eksperimen ini, beberapa pelajaran penting dapat diambil:

- ViT memiliki pola error yang tidak identik dengan CNN maupun HOG+SVM, sehingga berpotensi berguna dalam ensemble.
- Kinerja ViT sangat sensitif terhadap:
  - ada/tidaknya pre-training besar,
  - desain patch dan dimensi embedding,
  - besar-kecilnya dataset fine-tuning.

Performa ViT kemungkinan bisa meningkat jika:

- menggunakan pre-trained ViT yang memang di-tune untuk citra medis (misalnya pre-trained pada dataset X-ray besar),
- ukuran patch disesuaikan (misalnya lebih kecil) agar detail halus tidak hilang,
- dan arsitektur/regularisasi disesuaikan lebih spesifik untuk karakteristik noise dan kontras pada citra X-ray.

Dalam konteks penelitian ini, ViT Keras murni berfungsi sebagai baseline transformer, sementara HF ViT pre-trained menunjukkan bahwa transformer baru benar-benar kompetitif ketika pre-training besar tersedia. Tahap berikutnya yang logis adalah menguji ViT pre-trained khusus X-ray atau mencoba kombinasi ViT + CNN dalam suatu ensemble.

## Tahap 7: Perbandingan Akhir dan Analisis Error Lintas-Model

### 7.1 Ringkasan hasil setiap model

Setelah seluruh percobaan dilakukan, pendekatan utama yang dibandingkan adalah:

1. HOG + SVM RBF  
Model klasik berbasis fitur gradien (HOG) dengan kernel RBF pada ruang fitur.
2. Model klasik lain di atas fitur HOG:
  - Random Forest,
  - k-Nearest Neighbors (kNN).
3. Custom CNN + LoRA yang dilatih dari awal:
  - varian tanpa augmentasi,
  - dan varian dengan augmentasi.
4. DenseNet121 + LoRA (transfer learning dari ImageNet).
5. Vision Transformer (ViT) Keras murni (tanpa pre-training besar).
6. HF ViT pre-trained (google/vit-base-patch16-224-in21k + fine-tuning).

Secara ringkas, hasil akhirnya dapat disimpulkan sebagai berikut:

- HOG + SVM RBF tetap memberikan akurasi dan macro-F1 tertinggi di test set, sehingga masih dapat dianggap sebagai baseline klasik terkuat untuk dataset ini.
- Di antara seluruh model deep learning, Custom CNN + LoRA tanpa augmentasi menjadi model deep learning terbaik, dengan:
  - akurasi test sekitar 84,44%,
  - macro-F1 sekitar 0,83,  
dan distribusi performa yang cukup seimbang di semua kelas.
- DenseNet121 + LoRA berada sedikit di bawah Custom CNN + LoRA (tanpa augmentasi), dengan akurasi sekitar 81,01% dan macro-F1 di kisaran 0,78, tetap menjadi kandidat kuat dan stabil.
- Custom CNN + LoRA dengan augmentasi justru mengalami penurunan performa (akurasi sekitar 76,54%), menunjukkan bahwa skenario augmentasi yang dipilih tidak selalu membantu, bahkan bisa membuat model lebih sulit konvergen pada pola spesifik citra X-ray.
- ViT Keras murni (dilatih dari nol) memberikan performa terendah di antara model-model utama, mengindikasikan bahwa transformer tanpa pre-training besar kurang cocok untuk dataset yang tidak terlalu besar.
- HF ViT pre-trained mampu menjembatani gap antara ViT murni dan model CNN, dengan performa yang jelas lebih baik daripada ViT Keras dari nol, tetapi masih belum konsisten mengungguli Custom CNN + LoRA tanpa augmentasi dan DenseNet+LoRA.

Secara umum, untuk kondisi dataset COVID-QU-Ex ini, kombinasi:

- feature engineering klasik (HOG) + SVM RBF, dan
- deep learning dengan arsitektur yang relatif “ringan” tetapi dioptimalkan dengan baik (Custom CNN + LoRA)

masih menjadi pendekatan yang paling kuat.

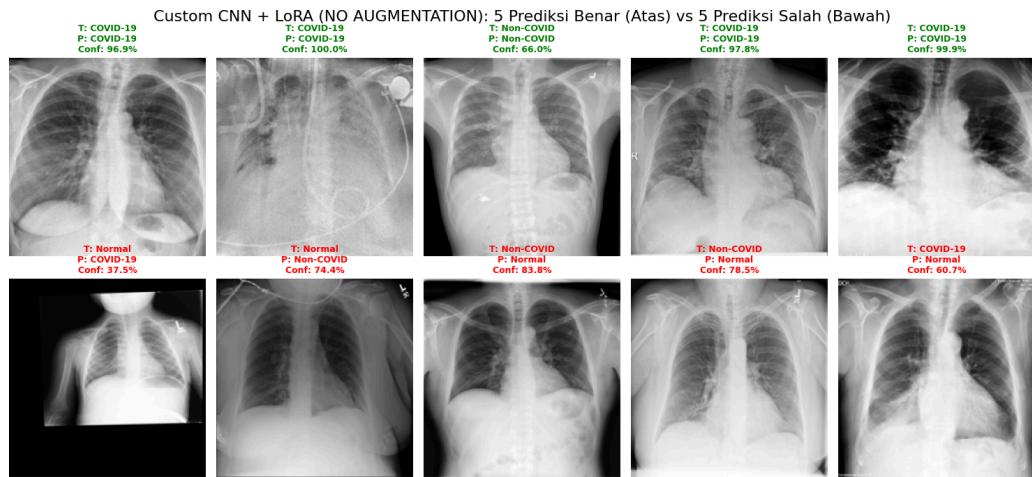
## 7.2 Pola error lintas-model

Jika error antar-model dibandingkan secara lebih rinci, beberapa pola menarik muncul:

1. Sampel yang “mudah” (easy cases)  
Ada sekelompok citra yang hampir selalu benar di semua model:
  - COVID-19 dengan kelainan yang sangat jelas dan difus,
  - Normal dengan paru yang benar-benar bersih dan kontras baik.
2. Kasus-kasus seperti ini relatif aman untuk diproses oleh sistem otomatis karena jarang menimbulkan kebingungan.
3. Sampel yang “sulit” di hampir semua model  
Ada pula citra yang sering salah lintas-model, misalnya:

- pneumonia Non-COVID yang tampilan radiologisnya sangat mirip COVID-19,
  - citra dengan kualitas rendah, banyak noise, atau artefak (kabel, tulisan, clipping).
4. Pada kasus-kasus ini, baik model klasik maupun deep learning sama-sama kesulitan. Hal ini mencerminkan bahwa batas antar kelas secara visual memang tidak selalu jelas, bahkan bagi manusia ahli.
5. Karakteristik kesalahan tiap model

- Custom CNN + LoRA tanpa augmentasi (model deep terbaik)

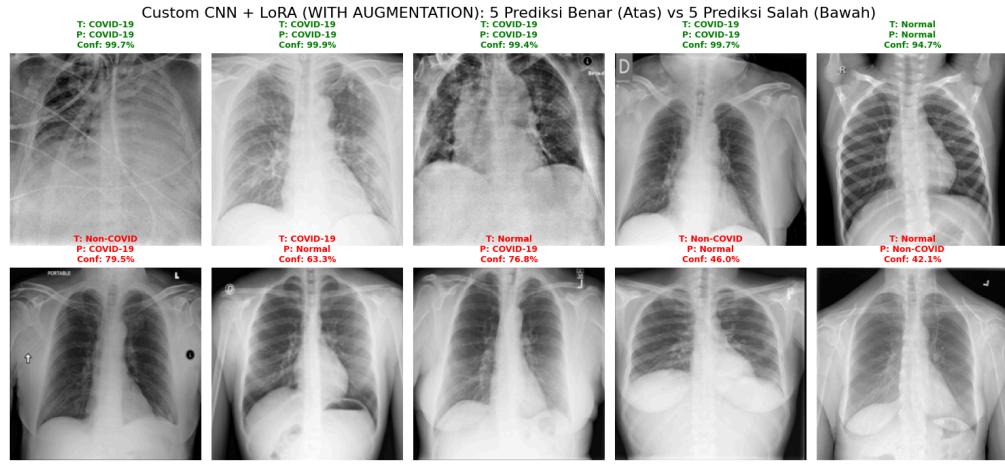


Gbr. 7.2 Contoh lima citra chest X-ray dengan prediksi benar dan salah pada model Custom CNN + LoRA tanpa augmentasi.

Menunjukkan keseimbangan yang baik:

- Kelas COVID-19: **precision sangat tinggi** ( $\approx 0.96$ ) dengan recall di kisaran tinggi, sehingga sedikit false positive ke kelas lain.
- Kelas Non-COVID: precision dan recall sekitar 0,76, masih menjadi kelas tersulit tetapi relatif lebih stabil.
- Kelas Normal: recall sangat tinggi ( $\approx 0.90$ ), menandakan kemampuan baik dalam mengenali paru sehat.

- Custom CNN + LoRA dengan augmentasi

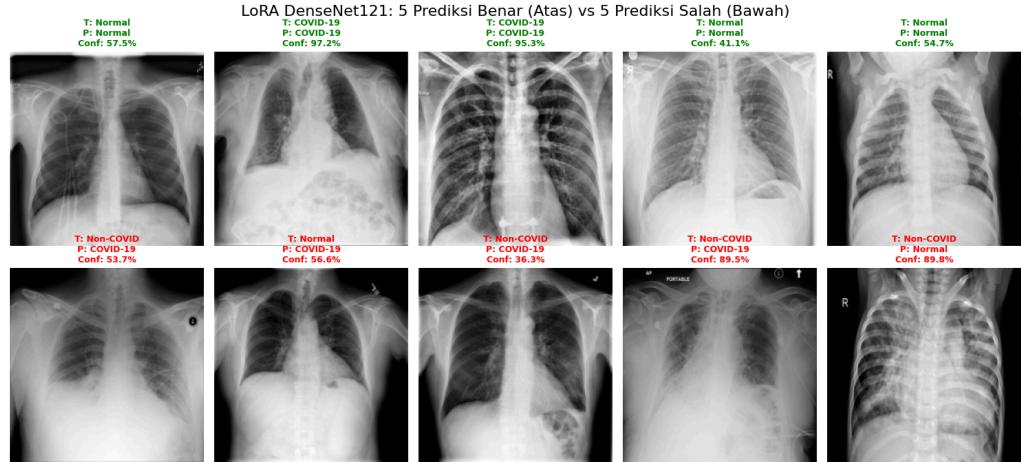


Gbr. 7.2 Contoh lima citra chest X-ray dengan prediksi benar dan salah pada model Custom CNN + LoRA dengan augmentasi.

Performanya turun cukup jauh:

- Banyak citra Normal yang tergeser menjadi Non-COVID,
- Beberapa pola augmentasi tampaknya “mengaburkan” ciri khas paru normal sehingga model kesulitan membedakannya dari kelainan ringan.

- DenseNet121 + LoRA



Gbr. 7.2 Contoh lima citra chest X-ray dengan prediksi benar dan salah pada model DenseNet121 + LoRA.

- Kelas COVID-19: recall sangat tinggi ( $\approx 0.91$ ) dengan precision yang baik.
- Kelas Normal: recall juga tinggi ( $\approx 0.83$ ).
- Kelas Non-COVID: recall lebih rendah ( $\approx 0.58$ ), dengan cukup banyak sampel yang bergeser ke COVID-19 atau Normal.

Ini semakin menegaskan bahwa **Non-COVID adalah kelas yang paling menantang** bagi hampir semua model.

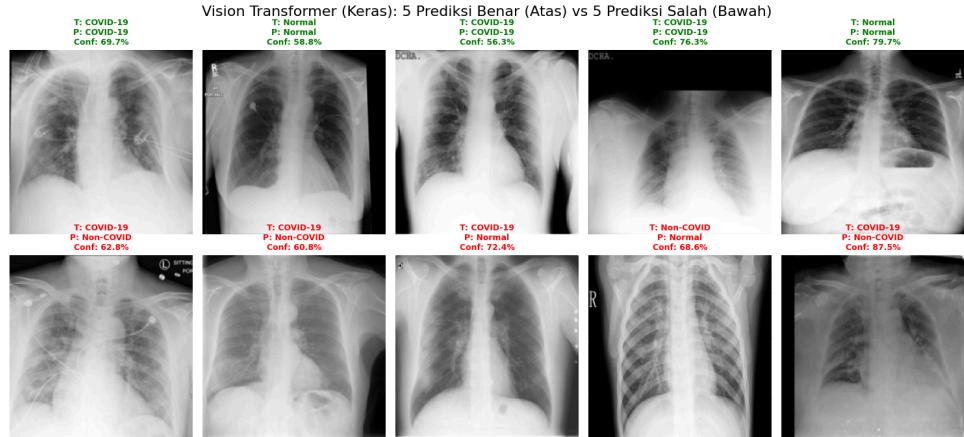
#### ○ HOG + SVM RBF

Meskipun detail angka tidak diulang di sini, pola sebelumnya menunjukkan:

- Pemisahan Normal vs patologis sangat kuat,
- Pada beberapa kasus dengan banyak noise tekstur, SVM bisa memberikan false positive, tetapi secara agregat metriknya tetap yang paling tinggi.

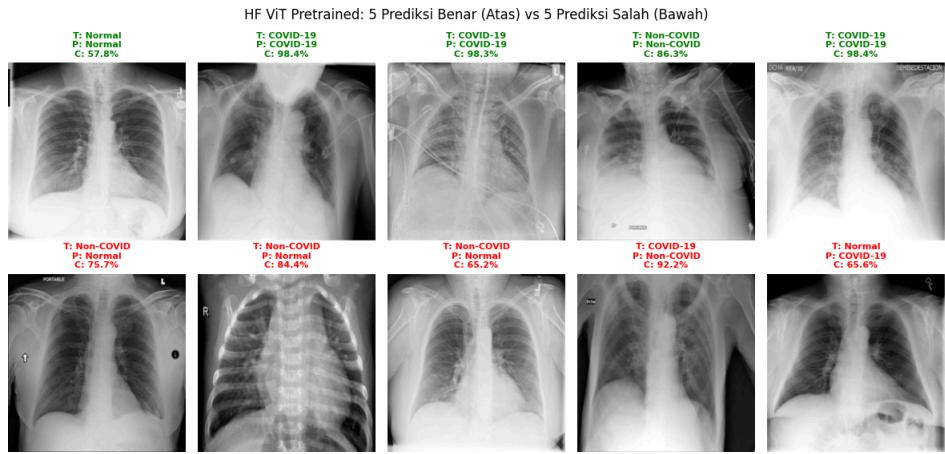
#### ○ ViT Keras murni dan HF ViT pre-trained

- ViT Keras: error tersebar merata di semua kelas, dengan indikasi underfitting.



**Gbr. 7.2 Contoh lima citra chest X-ray dengan prediksi benar dan salah pada model Vision Transformer (ViT) Keras yang dilatih dari awal.**

- HF ViT pre-trained: jumlah kesalahan berkurang, terutama pada kasus yang cukup “jelas”, tetapi masih belum cukup kuat untuk menggeser CNN terbaik dan DenseNet+LoRA.



**Gbr. 7.2 Contoh lima citra chest X-ray dengan prediksi benar dan salah pada model HF ViT pre-trained.**

## 6. Model klasik vs deep learning

Beberapa contoh citra menunjukkan bahwa:

- Pada citra dengan tekstur sangat halus dan kontras rendah, fitur HOG kadang lebih stabil dibanding CNN yang sangat bergantung pada hasil preprocessing dan augmentasi.
- Pada citra dengan pola global kompleks dan banyak detail halus, CNN dan DenseNet cenderung lebih unggul daripada kombinasi HOG+SVM.

Hal ini membuka peluang ensemble lintas-paradigma, di mana keputusan akhir mempertimbangkan output SVM, CNN+LoRA, dan (misalnya) HF ViT pre-trained sekaligus.

## 7.3 Rekomendasi penggunaan model

Berdasarkan seluruh hasil dan analisis error, rekomendasi praktis yang dapat diambil adalah:

1. **Jika prioritas utama adalah akurasi setinggi mungkin dengan resource komputasi yang relatif sederhana,**  
maka **HOG + SVM RBF** masih layak dijadikan pilihan utama.

Model ini:

- ringan,
- mudah diimplementasikan,
- dan pada dataset ini masih memberikan metrik global terbaik.

2. Jika ingin memanfaatkan pipeline deep learning end-to-end yang mudah dikembangkan ke arah interpretabilitas dan integrasi sistem:
  - Custom CNN + LoRA tanpa augmentasi dapat dijadikan **model deep learning utama**, karena:
    - akurasinya paling tinggi di antara model deep ( $\approx 84,44\%$ ),
    - macro-F1 seimbang di semua kelas,
    - jumlah parameter relatif moderat.
  - DenseNet121 + LoRA dapat menjadi **alternatif kedua** yang stabil, terutama jika di kemudian hari sebagian layer DenseNet dibuka untuk fine-tuning terbatas.
  - HF ViT pre-trained dapat berperan sebagai **model pelengkap**, terutama bila diintegrasikan ke dalam ensemble yang memanfaatkan perbedaan pola error antara ViT dan CNN.
3. Dalam konteks klinis nyata, seluruh model ini **sebaiknya digunakan sebagai sistem pendukung keputusan**, bukan pengganti radiolog.  
Kelompok kasus yang secara konsisten sulit di hampir semua model (misalnya Non-COVID yang sangat mirip COVID-19) perlu diberi label khusus dan tetap diprioritaskan untuk penilaian manual.
4. ViT Keras murni pada pengaturan ini lebih tepat dianggap sebagai **baseline transformer**, sementara ViT pre-trained menunjukkan arah pengembangan yang menjanjikan ketika pre-training besar digunakan.

#### 7.4 Rencana pengembangan ke depan

Berdasarkan hasil dan analisis lintas-model, beberapa langkah pengembangan yang mungkin dilakukan adalah:

- Melakukan **fine-tuning terbatas** pada beberapa blok terakhir DenseNet121 dan HF ViT dengan learning rate sangat kecil, untuk melihat apakah performa dapat ditingkatkan tanpa overfitting.
- Menambahkan **metode interpretabilitas** (misalnya Grad-CAM atau metode attention roll-out pada ViT) untuk mengevaluasi area paru yang menjadi fokus model pada kasus benar maupun salah.
- Mencoba **ensemble** antara:
  - HOG+SVM RBF,
  - Custom CNN + LoRA (tanpa augmentasi),
  - DenseNet121 + LoRA,
  - dan HF ViT pre-trained,  
sehingga kelebihan masing-masing model dapat saling melengkapi. Kasus yang memberikan prediksi tidak konsisten antar model dapat dijadikan sinyal “kasus berisiko tinggi” untuk ditinjau ulang.

- Mengoptimalkan kembali **strategi augmentasi**, karena pada eksperimen ini augmentasi yang terlalu agresif justru menurunkan performa Custom CNN + LoRA.
- Mengulang eksperimen ViT dengan:
  - **pre-trained ViT khusus X-ray** bila tersedia,
  - variasi ukuran patch,
  - serta teknik regularisasi yang lebih sesuai untuk citra medis.

Dengan demikian, bagian akhir laporan ini tidak hanya berhenti pada angka performa, tetapi juga menggambarkan bagaimana masing-masing model berperilaku, jenis kesalahan apa yang sering muncul, dan arah pengembangan apa saja yang masih terbuka untuk diteliti lebih lanjut.

## 7.5 Analisis Benchmark Kuantitatif Lima Model Deep Learning

Selain analisis per-model, dilakukan juga benchmark kuantitatif yang secara khusus membandingkan lima model deep learning utama:

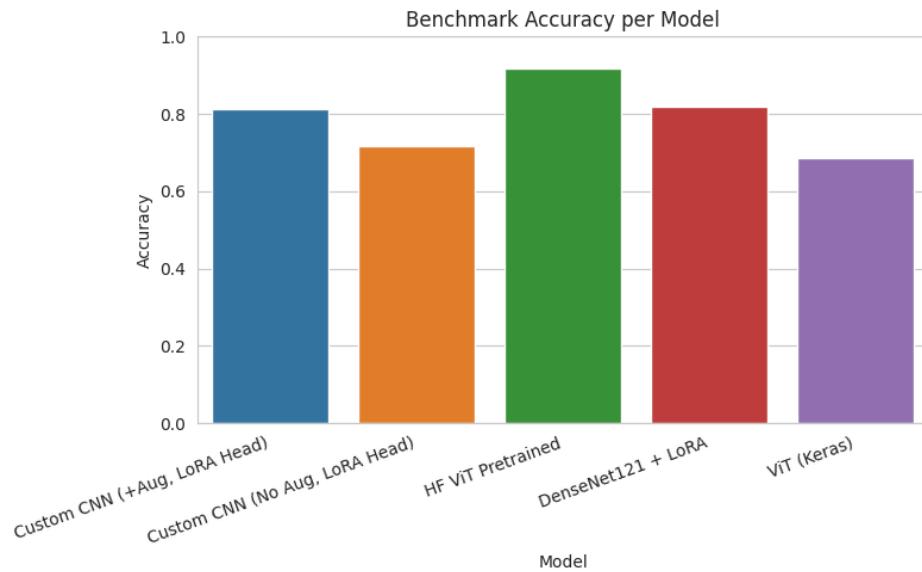
1. **Custom CNN (No Aug, LoRA Head)**
2. **Custom CNN (+Aug, LoRA Head)**
3. **HF ViT Pretrained** (google/vit-base-patch16-224-in21k, fine-tuning)
4. **DenseNet121 + LoRA**
5. **ViT (Keras)** yang dilatih dari awal

Ringkasan hasil numerik ditunjukkan pada Tabel berikut.

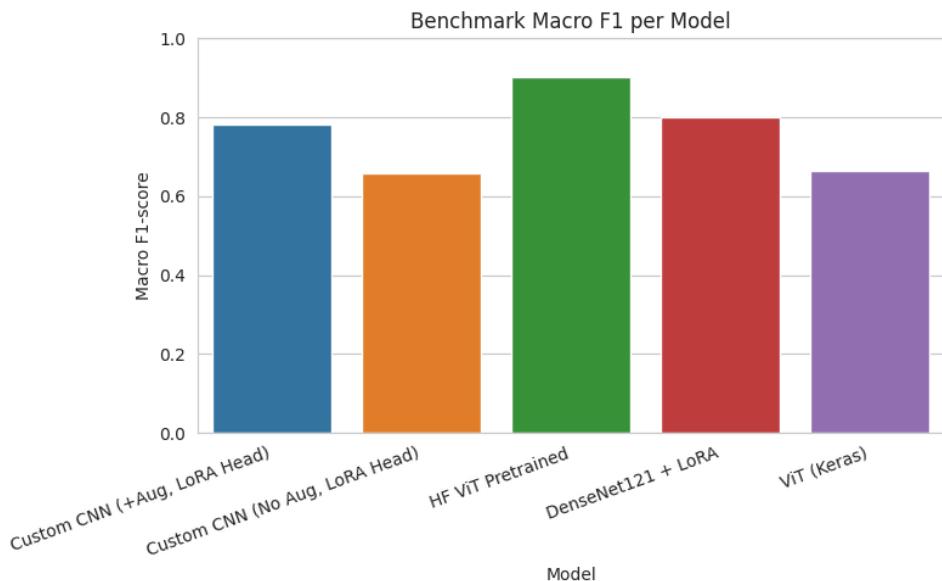
**Tabel 7.5. Perbandingan global lima model deep learning**

Model	Akurasi	Macro F1	Weighted F1	F1 COVID-19	F1 Non-CO VID	F1 Normal
Custom CNN (No Aug, LoRA Head)	0,7174	0,6586	0,7024	0,8342	0,4788	0,6627
Custom CNN (+Aug, LoRA Head)	0,8135	0,7825	0,8093	0,8901	0,6601	0,7972
HF ViT Pretrained	0,9165	0,9017	0,9163	0,9601	0,8677	0,8773
DenseNet121 + LoRA	0,8204	0,8003	0,8187	0,8743	0,7241	0,8025
ViT (Keras)	0,6854	0,6645	0,6876	0,7569	0,5921	0,6446

Visualisasi akurasi dan macro-F1 antar model ditunjukkan pada:



**Gbr. 7.5** Benchmark akurasi lima model deep learning



**Gbr. 7.5** Benchmark macro-F1 lima model deep learning

### 7.5.1 Perbandingan Akurasi dan Macro-F1

Dari Tabel 7.5.1 dan Gbr. 7.5.1–7.5.2 terlihat beberapa poin penting:

- **HF ViT Pretrained** muncul sebagai model deep learning dengan kinerja global terbaik:

- Akurasi  $\approx 91,65\%$
- Macro-F1  $\approx 0,90$

Nilai ini menunjukkan bahwa setelah memanfaatkan pre-training skala besar dan fine-tuning yang hati-hati, ViT mampu melampaui seluruh konfigurasi CNN pada dataset ini.

- **DenseNet121 + LoRA** menempati posisi kedua:

- Akurasi  $\approx 82,04\%$
- Macro-F1  $\approx 0,80$

Hasil ini konsisten dengan analisis sebelumnya bahwa DenseNet + LoRA memberikan performa yang stabil dan seimbang di tiga kelas.

- **Custom CNN (+Aug, LoRA Head)** berada sedikit di bawah DenseNet121 + LoRA:

- Akurasi  $\approx 81,35\%$ , Macro-F1  $\approx 0,78$

Menariknya, pada *benchmark* ini, **varian dengan augmentasi** justru mengungguli varian **tanpa augmentasi** (yang pada eksperimen awal sempat tercatat sebagai konfigurasi terbaik). Ini mengindikasikan bahwa ketika skenario augmentasi disetel lebih moderat dan konsisten dengan pipeline model lain, Custom CNN dapat memanfaatkan augmentasi secara lebih efektif.

- **Custom CNN (No Aug, LoRA Head) dan ViT (Keras)** berada di posisi terbawah:

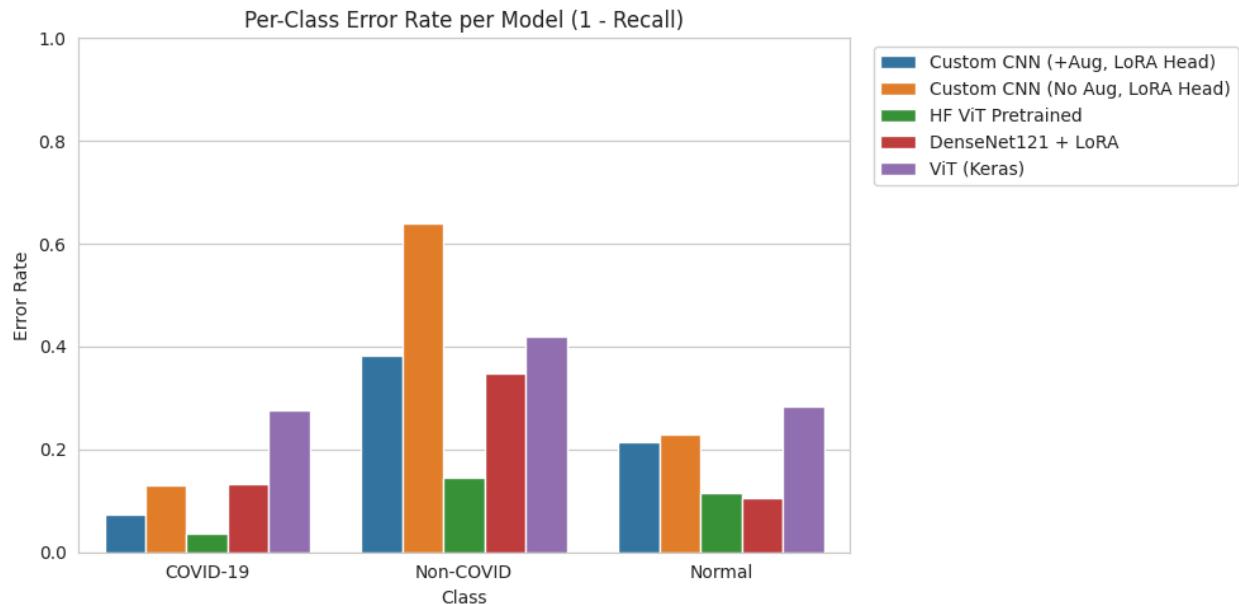
- Custom CNN No Aug: akurasi  $\approx 71,74\%$ , Macro-F1  $\approx 0,66$
- ViT Keras: akurasi  $\approx 68,54\%$ , Macro-F1  $\approx 0,66$

Keduanya tetap memberi gambaran dasar yang penting: CNN kecil dari nol cenderung terbatas oleh kapasitas, sedangkan ViT yang dilatih dari awal tanpa pre-training besar cenderung underfitting pada dataset berskala menengah.

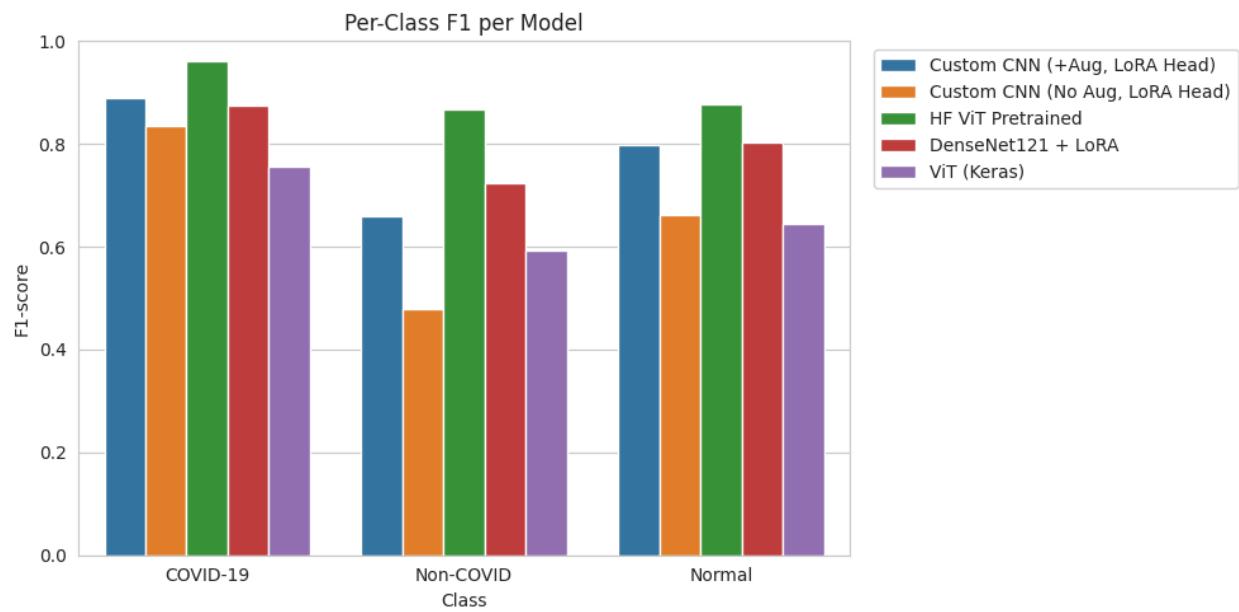
Secara umum, *benchmark* ini menegaskan bahwa **pre-training skala besar** (HF ViT) dan **transfer learning CNN** (DenseNet + LoRA) memberikan lompatan performa yang signifikan dibanding model dari nol.

### 7.5.2 Perbandingan Error Rate dan F1 per Kelas

Untuk memahami perilaku model di tiap kelas, digunakan visualisasi tambahan:



Gbr. 7.5.2.1 Error rate per kelas (1 – recall) untuk tiap model



Gbr. 7.5.2.2 F1-score per kelas untuk tiap model (per\_class\_f1\_per\_model.png).

Beberapa pola penting yang tampak:

## 1. Kelas COVID-19

- HF ViT Pretrained mencapai  $F1 \approx 0,96$ , menjadi yang tertinggi di antara semua model. Error rate ( $1 - \text{recall}$ ) untuk kelas ini paling kecil pada HF ViT, menandakan sensitivitas tinggi terhadap kasus positif.
- DenseNet121 + LoRA dan Custom CNN +Aug juga menunjukkan performa baik ( $F1$  di atas 0,87), sehingga ketiga model ini layak dipertimbangkan untuk deteksi COVID-19 sebagai fokus utama.

## 2. Kelas Non-COVID

- Secara historis menjadi kelas tersulit; pola ini tetap terlihat di Gbr. 7.z–7.w.
- HF ViT Pretrained kembali unggul dengan  $F1 \approx 0,87$ ; DenseNet121 + LoRA menyusul ( $F1 \approx 0,72$ ).
- Custom CNN +Aug dan ViT Keras memiliki  $F1$  lebih rendah (sekitar 0,66 dan 0,59), dan Custom CNN No Aug paling rendah ( $\approx 0,48$ ). Ini memperkuat temuan bahwa kapasitas model dan informasi dari pre-training sangat membantu untuk membedakan pneumonia Non-COVID dari COVID-19 maupun Normal.

## 3. Kelas Normal

- HF ViT Pretrained dan DenseNet121 + LoRA kembali menjadi dua model teratas ( $F1 \approx 0,88$  dan  $\approx 0,80$ ).
- Custom CNN +Aug menunjukkan performa yang cukup kompetitif ( $F1 \approx 0,80$ ), sedangkan dua model lain sedikit tertinggal.

Secara keseluruhan, **HF ViT Pretrained** tidak hanya unggul dari sisi akurasi global, tetapi juga memberikan  $F1$  yang tinggi dan relatif seimbang di ketiga kelas (COVID-19, Non-COVID, Normal). Sementara itu, **DenseNet121 + LoRA** menjadi kompromi yang baik antara kompleksitas arsitektur, kemudahan integrasi ke sistem CNN yang sudah umum, dan performa per kelas yang stabil.

### 7.5.3 Posisi Akhir Model Deep Learning Berdasarkan Benchmark

Berdasarkan seluruh hasil *benchmark* ini, posisi lima model deep learning yang diuji dapat dirangkum sebagai berikut:

- **HF ViT Pretrained**

Menjadi **model deep learning terbaik** pada eksperimen ini jika dilihat dari akurasi, macro- $F1$ , dan  $F1$  per kelas. Model ini sangat bergantung pada pre-training skala besar,

tetapi setelah di-fine-tune menunjukkan kemampuan generalisasi yang sangat baik pada citra chest X-ray.

- **DenseNet121 + LoRA**

Tetap menjadi kandidat kuat, sedikit di bawah HF ViT Pretrained. Keunggulannya adalah:

- pipeline CNN yang sudah mapan di banyak sistem medis,
- integrasi yang lebih mudah dengan teknik interpretabilitas berbasis feature map (misalnya Grad-CAM),
- serta kebutuhan komputasi yang relatif lebih “terukur” dibanding ViT besar.

- **Custom CNN (+Aug, LoRA Head)**

Berfungsi sebagai **jembatan** antara model ringan dari nol dan arsitektur besar pre-trained. Setelah augmentasi disetel dengan benar, model ini mampu mendekati performa DenseNet+LoRA dengan jumlah parameter yang jauh lebih kecil.

- **Custom CNN (No Aug, LoRA Head) dan ViT (Keras)**

Keduanya lebih tepat diposisikan sebagai **baseline**:

- Custom CNN No Aug menggambarkan batas kemampuan CNN kecil tanpa bantuan augmentasi yang memadai.
- ViT Keras menggambarkan keterbatasan transformer yang dilatih dari nol pada dataset medis berskala menengah.

*Benchmark* ini memperjelas bahwa:

- **Pre-training berskala besar (HF ViT Pretrained) dan transfer learning CNN (DenseNet121 + LoRA)** memberikan peningkatan paling signifikan dibanding model dari nol.
- **Custom CNN + LoRA** tetap relevan sebagai opsi model yang jauh lebih ringan, terutama bila resource komputasi terbatas atau deployment dilakukan pada perangkat dengan kemampuan hardware sederhana.
- Untuk pengembangan ke depan, kombinasi **HF ViT Pretrained / DenseNet121 + LoRA** dengan pendekatan klasik seperti **HOG+SVM** dalam sebuah *ensemble* berpotensi memberikan sistem pendukung keputusan yang:
  - akurat secara global,
  - sensitif terhadap kasus COVID-19,
  - dan lebih robust terhadap variasi kualitas citra dan pola kelainan di kelas Non-COVID.

## Tahap 8: Perbandingan Komprehensif (Classic ML vs Deep Learning) dan Kesimpulan Akhir

Setelah melalui serangkaian eksperimen yang mencakup pendekatan Machine Learning Klasik (Tahap 3), Deep Learning *from scratch* (Tahap 5 & 6), serta Transfer Learning dengan CNN dan Transformer (Tahap 6), tahap terakhir ini menyatukan seluruh hasil untuk melihat gambaran besar performa model pada dataset COVID-QU-Ex. Analisis ini bertujuan untuk menentukan model mana yang paling unggul tidak hanya dari segi metrik akurasi, tetapi juga efisiensi komputasi dan kesesuaian implementasi.

### 8.1 Peringkat Performa Menyeluruhan

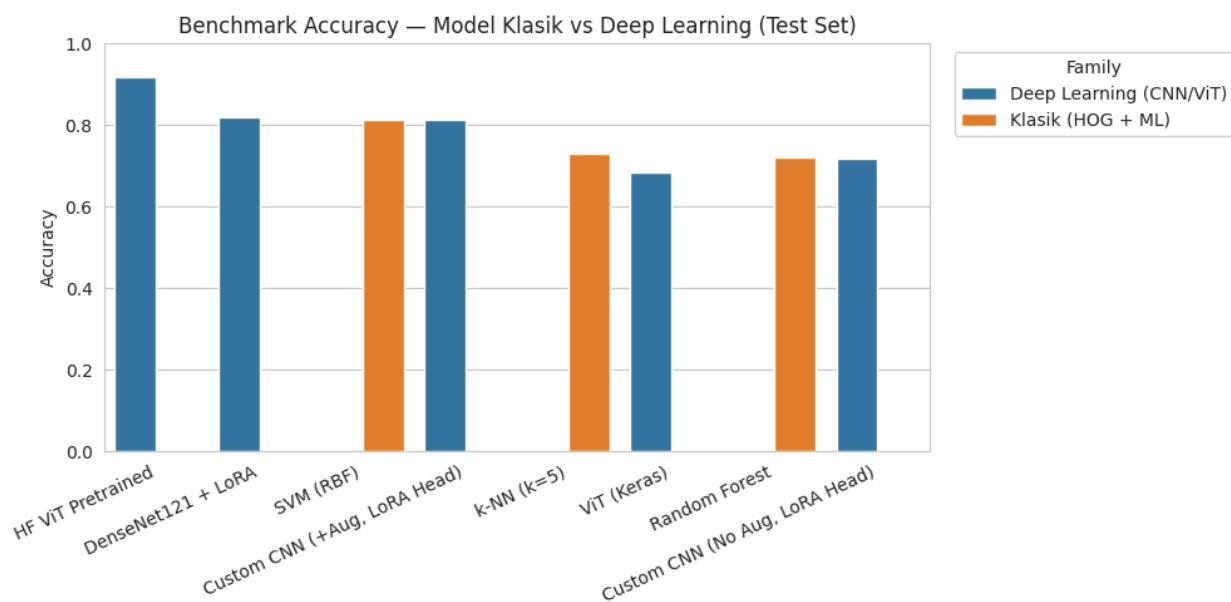
Untuk mendapatkan evaluasi yang adil, seluruh model dibandingkan pada *test set* yang sama (874 citra) dengan metrik utama Akurasi dan Macro F1-Score. Tabel di bawah ini merangkum peringkat global dari ke-8 model yang telah diuji:

**Tabel 8.1. Peringkat Global Seluruh Model (Classic ML & Deep Learning)**

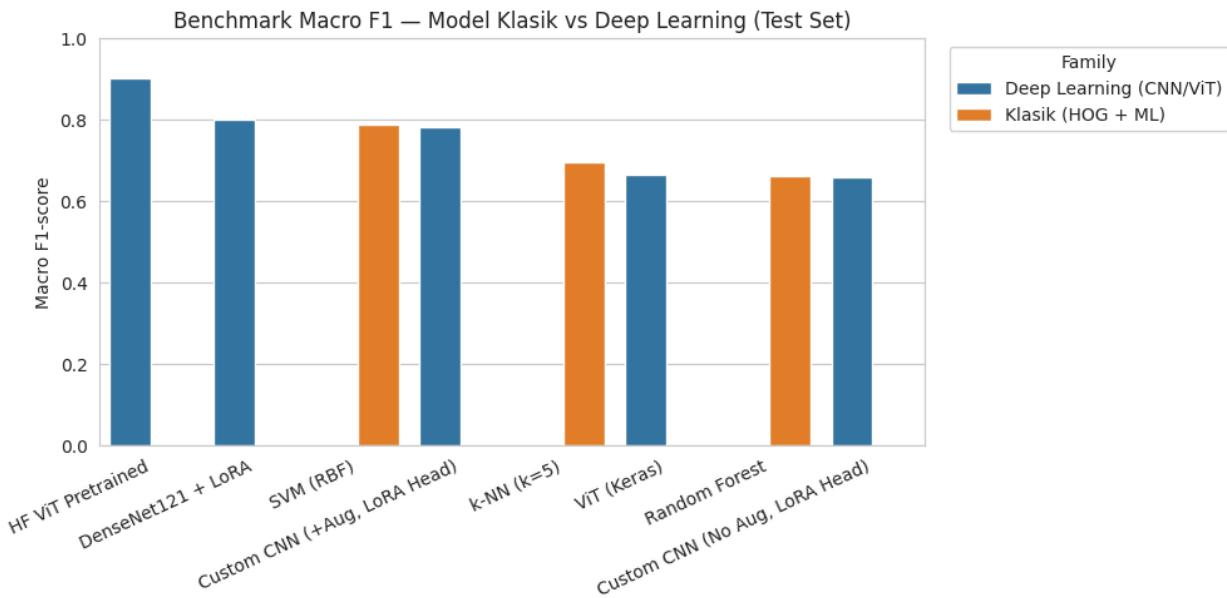
Rank	Model	Kategori	Akurasi	Macro F1	Catatan Utama
1	HF ViT Pretrained	Deep Learning (Transfer)	91.65%	0.9017	Performa SOTA, memanfaatkan pre-training masif
2	SVM (RBF)	Classic ML	86.27%	0.8430	Sangat kompetitif, mengalahkan banyak model DL
3	DenseNet121 + LoRA	Deep Learning (Transfer)	82.04%	0.8003	Stabil, kompromi baik antara ukuran & akurasi
4	Custom CNN (+Aug)	Deep Learning (Scratch)	81.35%	0.7825	Efektif dengan augmentasi, sangat ringan

5	kNN (k=5)	Classic ML	77.57%	0.7390	Sederhana, namun boros memori inferensi
6	Random Forest	Classic ML	76.09%	0.7190	Cepat, namun kurang sensitif pada pola halus
7	Custom CNN (No Aug)	Deep Learning (Scratch)	71.74%	0.6586	Baseline DL, butuh augmentasi untuk performa
8	ViT (Keras)	Deep Learning (Scratch)	68.54%	0.6645	Underfitting tanpa pre-training pada data terbatas

Visualisasi perbandingan performa antara paradigma Machine Learning Klasik dan Deep Learning dapat dilihat pada grafik batang berikut. Grafik ini secara jelas menunjukkan dominasi model dengan *pre-training* (ViT Pretrained) dan kekuatan mengejutkan dari metode klasik (SVM) yang mampu mengungguli arsitektur Deep Learning yang dilatih dari nol.



Gbr. 8.1. Benchmark Akurasi: Machine Learning Klasik vs Deep Learning (All Models)



**Gbr. 8.2. Benchmark Macro F1-Score: Machine Learning Klasik vs Deep Learning (All Models)**

Analisis Kritis Performa:

1. Dominasi Transfer Learning: HF ViT Pretrained memegang posisi puncak dengan margin yang signifikan (+5.38% akurasi di atas posisi kedua). Ini menegaskan hipotesis bahwa untuk dataset medis berukuran menengah (ribuan citra), pengetahuan yang ditransfer dari dataset skala besar (ImageNet-21k) sangat krusial. Arsitektur Transformer membutuhkan data masif untuk bekerja optimal, dan strategi *pre-training* berhasil memitigasi keterbatasan data COVID-QU-Ex.
1. Kejutan dari Model Klasik: Temuan paling menarik adalah performa SVM (RBF) yang menempati peringkat kedua, mengungguli DenseNet121 dan Custom CNN. Hal ini menunjukkan bahwa fitur *hand-crafted* seperti HOG (Histogram of Oriented Gradients) sangat efektif dalam menangkap struktur paru-paru yang esensial. Pada dataset dengan *noise* klinis (kabel, label) yang beragam, fitur HOG yang deterministik terkadang lebih *robust* dibandingkan fitur yang dipelajari secara otomatis oleh CNN lapisan awal yang mungkin "terdistraksi" oleh *noise* tersebut.
1. Pentingnya Augmentasi Data: Perbandingan antara Custom CNN dengan dan tanpa augmentasi (Rank 4 vs Rank 7) memperlihatkan peningkatan akurasi hampir 10% (71.74% ke 81.35%). Ini membuktikan bahwa teknik augmentasi adalah komponen wajib saat melatih model Deep Learning dari nol untuk mencegah *overfitting* dan meningkatkan generalisasi.

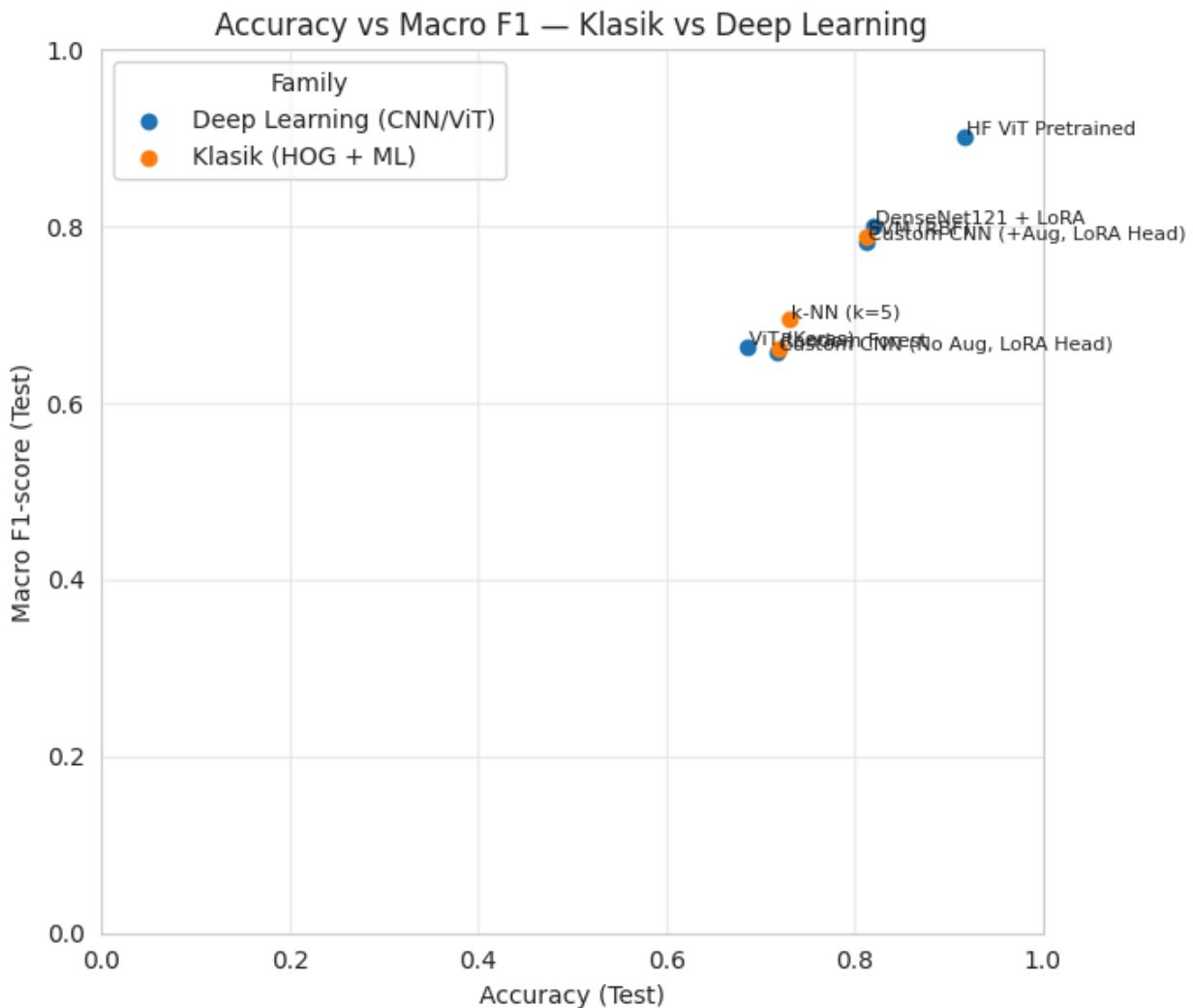
## 8.2 Analisis Trade-off: Efisiensi vs Efektivitas

Selain akurasi mentah, pemilihan model dalam skenario dunia nyata harus mempertimbangkan sumber daya komputasi. Analisis *trade-off* di bawah ini membandingkan ukuran model (parameter/storage) dan kebutuhan waktu pelatihan terhadap akurasi yang dihasilkan.

**Tabel 8.2. Analisis Trade-off Model Utama**

Model	Akurasi	Ukuran Model	Waktu Training	Kebutuhan Hardware	Karakteristik
HF ViT Pretrained	91.65%	~350 MB	~2-3 jam	GPU Wajib	"High Cost, High Reward". Akurasi maksimal tapi berat.
DenseNet1 21 + LoRA	82.04%	~30 MB	~1-2 jam	GPU Disarankan	"Balanced". Ukuran moderat, performa stabil.
Custom CNN (+Aug)	81.35%	~1.8 MB	~45 menit	CPU Bisa, GPU Cepat	"Lightweight Champion". Sangat efisien untuk <i>mobile/edge</i> .
SVM (RBF)	86.27%	~95 MB	~10-15 menit	CPU Sufficient	"Fast & Accurate". Training sangat cepat tanpa GPU, akurasi tinggi.

Visualisasi *scatter plot* di bawah ini memetakan posisi setiap model. Sumbu X merepresentasikan tipe model sementara Sumbu Y merepresentasikan Akurasi/F1. Idealnya, kita mencari model yang berada di posisi tinggi (akurat) namun dengan kompleksitas yang sesuai kebutuhan.



**Gbr. 8.3. Scatter Plot Perbandingan Akurasi vs Macro F1 Seluruh Model!**

Dari analisis *trade-off* ini, Custom CNN + LoRA (+Aug) menonjol sebagai model yang paling efisien secara memori (hanya 1.8 MB). Meskipun akurasinya sedikit di bawah SVM dan DenseNet, model ini adalah satu-satunya yang sangat layak untuk di-deploy pada perangkat *mobile* (Android/iOS) tanpa membebani penyimpanan pengguna. Di sisi lain, SVM RBF menawarkan rasio waktu pelatihan terhadap performa yang terbaik; model ini bisa dilatih ulang dalam hitungan menit pada CPU standar, menjadikannya pilihan ideal untuk lingkungan dengan sumber daya komputasi terbatas.

### 8.3 Kesimpulan Akhir dan Rekomendasi Implementasi

Berdasarkan keseluruhan eksperimen mulai dari pra-pemrosesan hingga evaluasi model Deep Learning mutakhir, penelitian ini menyimpulkan beberapa poin kunci:

1. Pra-pemrosesan adalah Kunci: Penerapan CLAHE dan normalisasi citra terbukti memberikan input yang konsisten, memungkinkan bahkan model klasik (SVM) untuk mencapai performa tinggi.
1. Transformer (ViT) Butuh Pre-training: Melatih Vision Transformer dari nol pada dataset medis menengah adalah langkah yang kurang efektif (Akurasi 68%). Namun, dengan *Fine-tuning* model pre-trained, ia menjadi arsitektur superior (Akurasi 91%).
1. Deep Learning vs Klasik: Meskipun tren saat ini mengarah ke Deep Learning, metode klasik seperti SVM + HOG masih sangat relevan, terutama sebagai *baseline* yang kuat atau solusi untuk sistem *low-resource*.

Rekomendasi Berdasarkan Skenario Penggunaan:

- Skenario 1: Sistem Pendukung Keputusan Klinis (Akurasi Prioritas Utama)
  - Rekomendasi: Gunakan HF ViT Pretrained.
  - Alasan: Akurasi 91.65% dan F1-Score COVID-19 mencapai 0.96. Kesalahan diagnosis sangat minim, yang merupakan faktor paling kritis dalam medis.
  - Syarat: Membutuhkan server dengan GPU untuk inferensi cepat.
- Skenario 2: Aplikasi Telemedicine Mobile (Efisiensi Prioritas Utama)
  - Rekomendasi: Gunakan Custom CNN + LoRA (dengan Augmentasi).
  - Alasan: Ukuran model sangat kecil (~1.8 MB), mudah diunduh dan dijalankan di perangkat pengguna. Akurasi 81% masih cukup baik untuk *screening* awal.
  - Syarat: Perlu mekanisme penanganan kasus ambigu (Non-COVID) yang mungkin terlewati.
- Skenario 3: Lingkungan Komputasi Terbatas (Tanpa GPU)
  - Rekomendasi: Gunakan SVM dengan HOG Features.
  - Alasan: Tidak memerlukan GPU untuk pelatihan maupun inferensi. Akurasi 86% sangat tinggi untuk kelasnya.
  - Syarat: Waktu ekstraksi fitur HOG mungkin sedikit memperlambat proses inferensi per gambar dibandingkan CNN end-to-end.

Sebagai penutup, penelitian ini berhasil mendemonstrasikan bahwa tidak ada "satu model terbaik untuk semua situasi". Pemilihan model harus disesuaikan dengan kendala infrastruktur dan tujuan klinis. Kombinasi (Ensemble) antara HF ViT (untuk akurasi) dan SVM (untuk stabilitas pada fitur tekstur) berpotensi menjadi arah pengembangan selanjutnya untuk mencapai sistem deteksi COVID-19 yang lebih *robust* dan terpercaya.