

Laporan Project - Pembelajaran Mesin

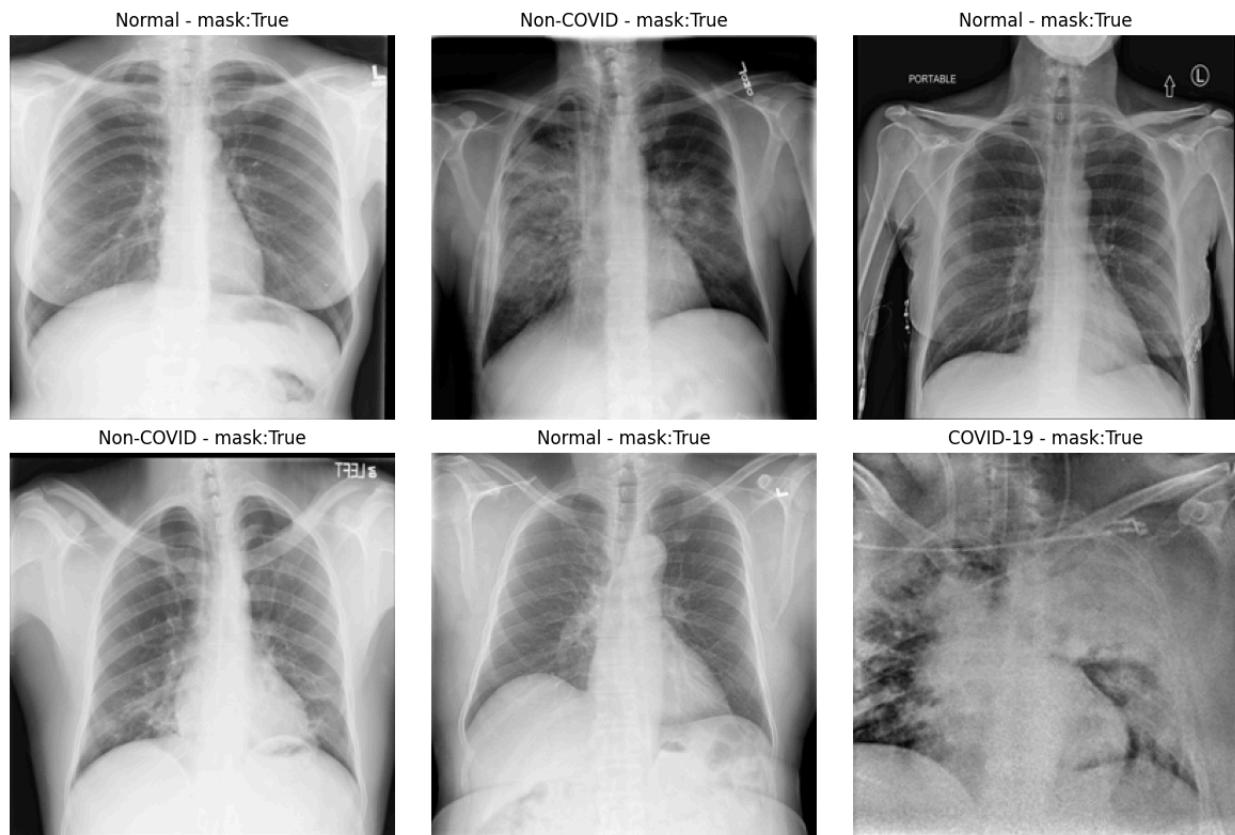
Anggota Kelompok :

202210370311060 - Muhammad Hisyam Kamil

202210370311449 - Elga Putri Tri Farma

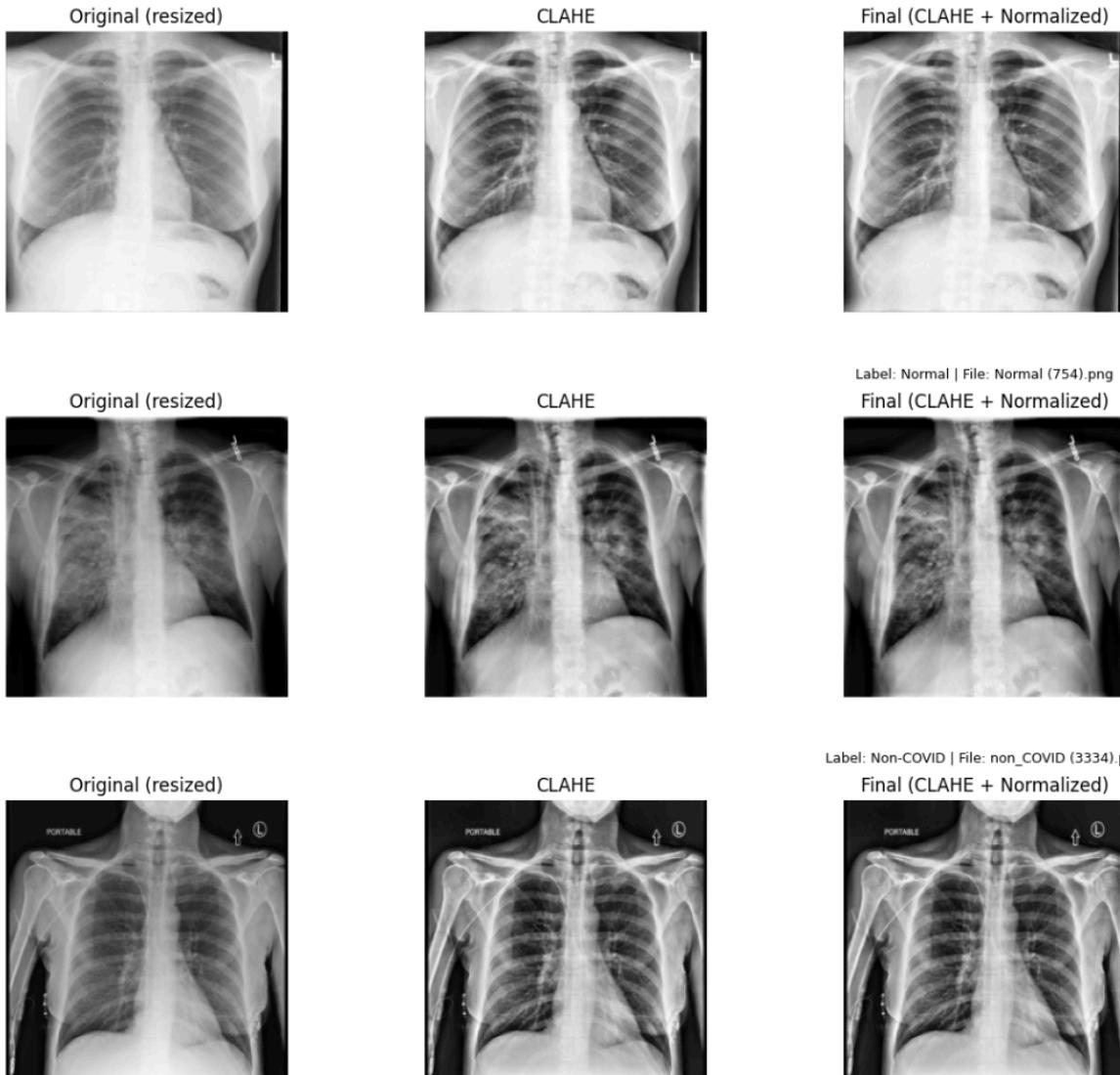
Tahap 1: Preprocessing Data

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 224×224 piksel, lalu ditingkatkan kontrasnya menggunakan CLAHE dengan pengaturan clip limit 2.0 dan tile grid 8×8 . 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% |
| Total | 5.826 | 100% |

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 |
| Total | 5.826 | 100% | - |

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 16×16 piksel, dan blok 2×2 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 |
| Dimensi HOG | 6.084 | 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 |
| 4.096 | 0,7669 |

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 |

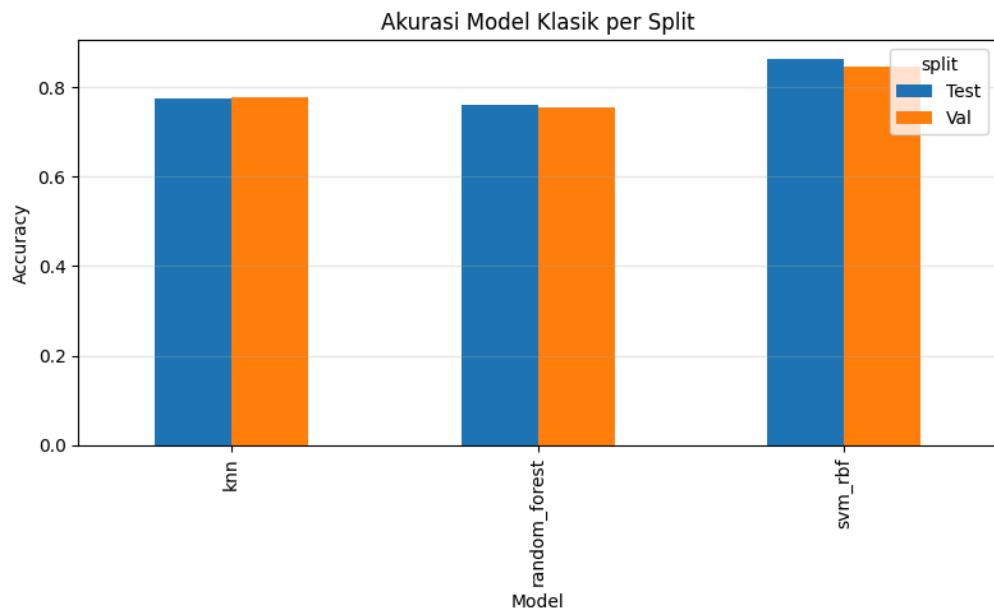
Tahap 3: Pelatihan Model Klasik

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.



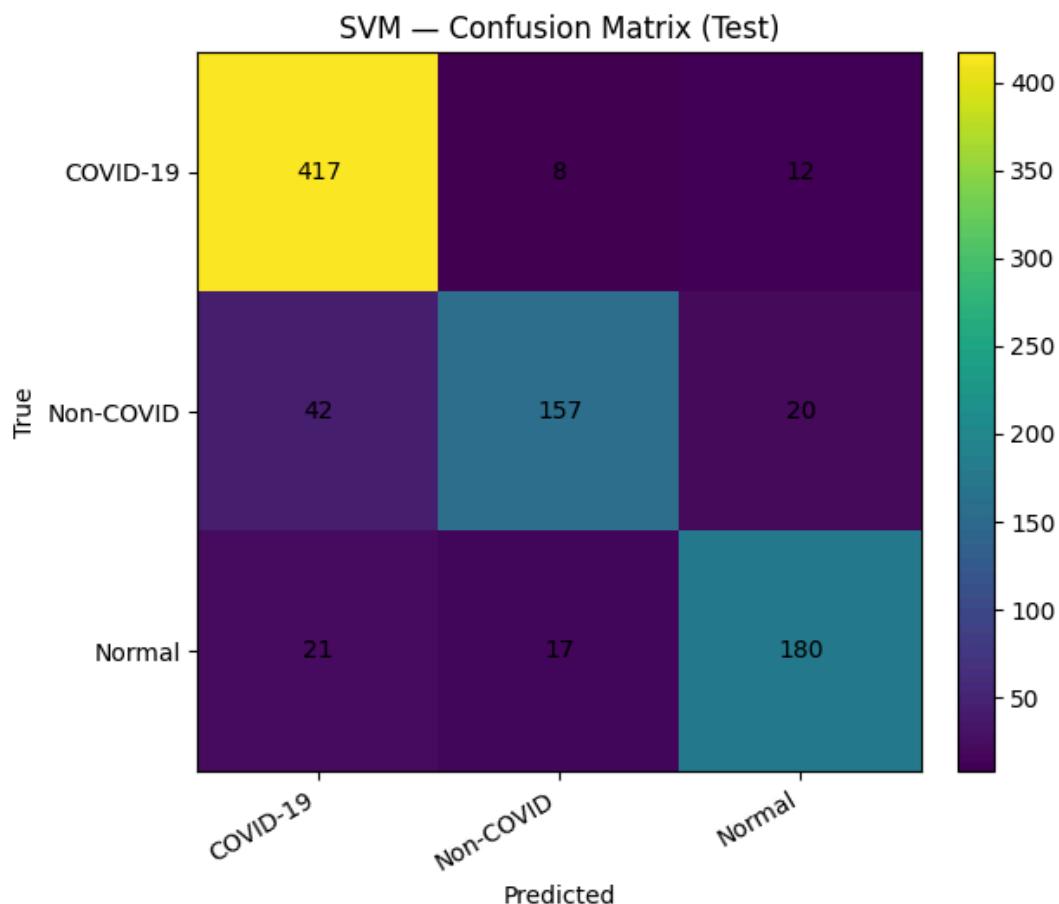
Gbr. 3.1. Perbandingan Akurasi Model Klasik per Split

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

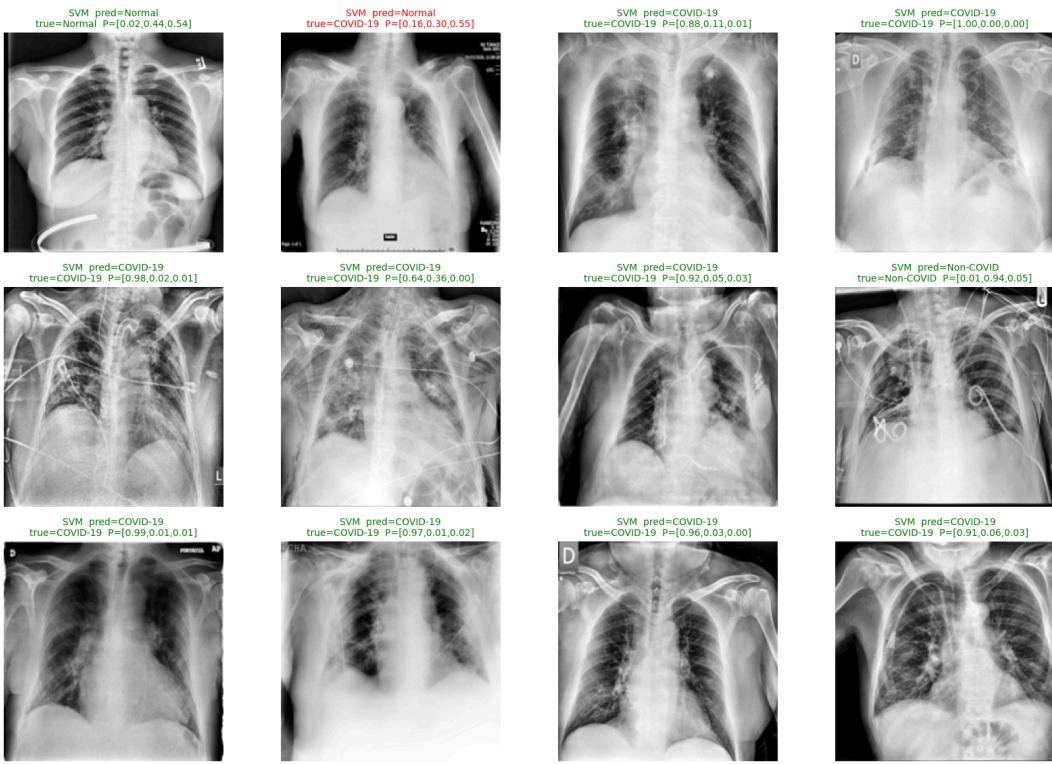
| Model | Split | Akurasi | Precision (macro) | Recall (macro) | F1 (macro) |
|--------------|-------|---------------|-------------------|----------------|---------------|
| SVM RBF | Val | 0,8478 | 0,8313 | 0,8178 | 0,8234 |
| SVM RBF | Test | 0,8627 | 0,8601 | 0,8323 | 0,8432 |
| RandomForest | Val | 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%.

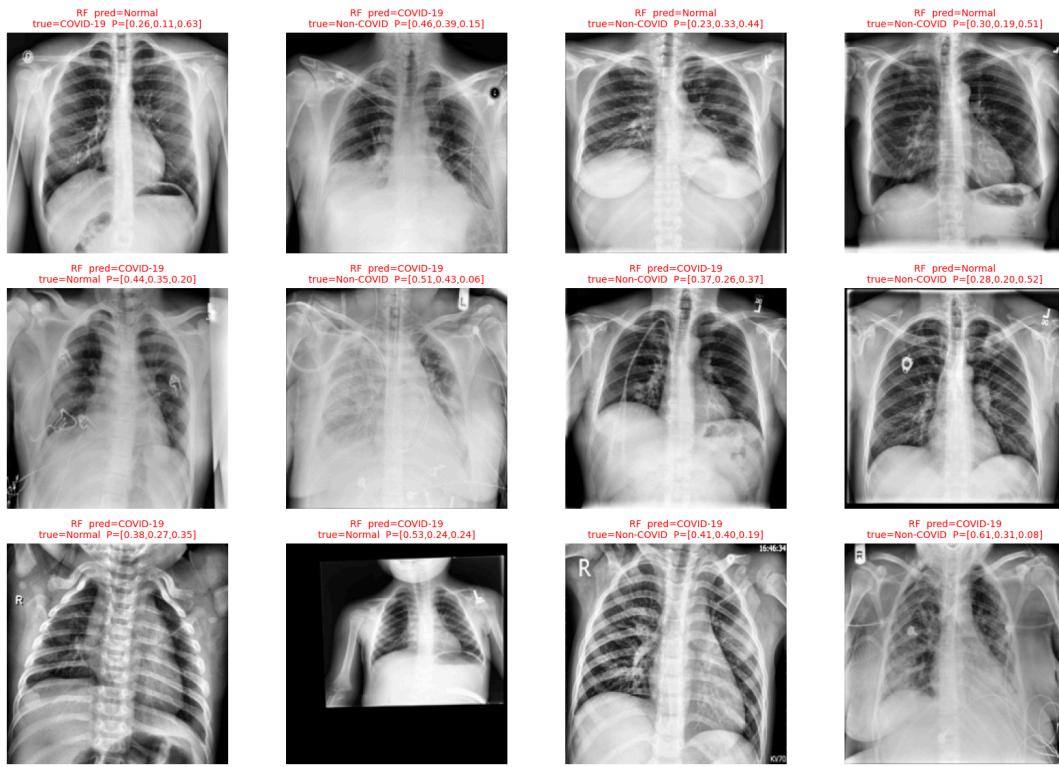


Gbr. 3.2. Confusion Matrix SVM on Test Dataset

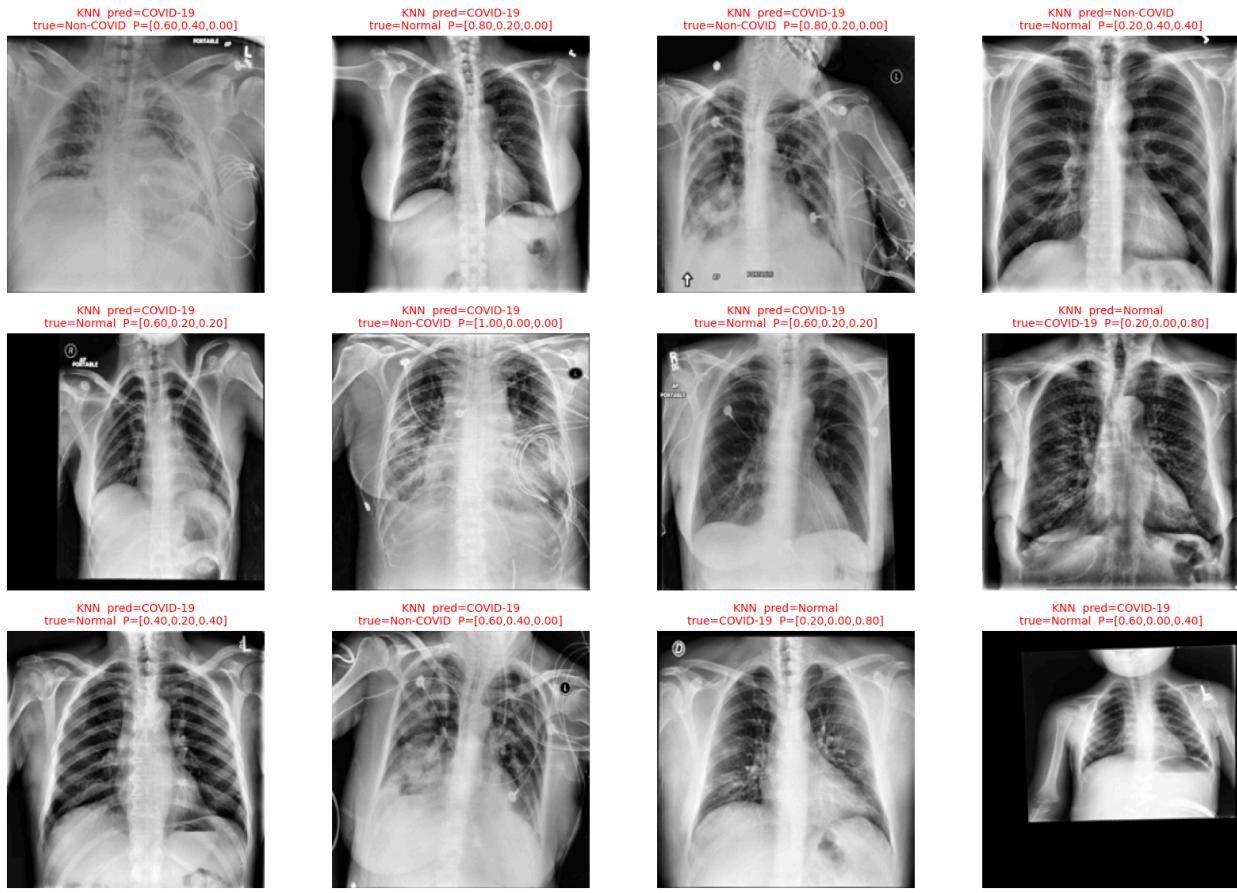
Kelas Non-COVID masih menjadi sumber kesalahan terbesar karena sebagian kasusnya didorong ke kelas COVID-19. Dari peninjauan visual, kesalahan seperti ini kerap terjadi pada citra yang menyertakan kabel, tube, atau artefak klinis lain yang memperkaya struktur tepi sehingga “terbaca” sebagai pola yang 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

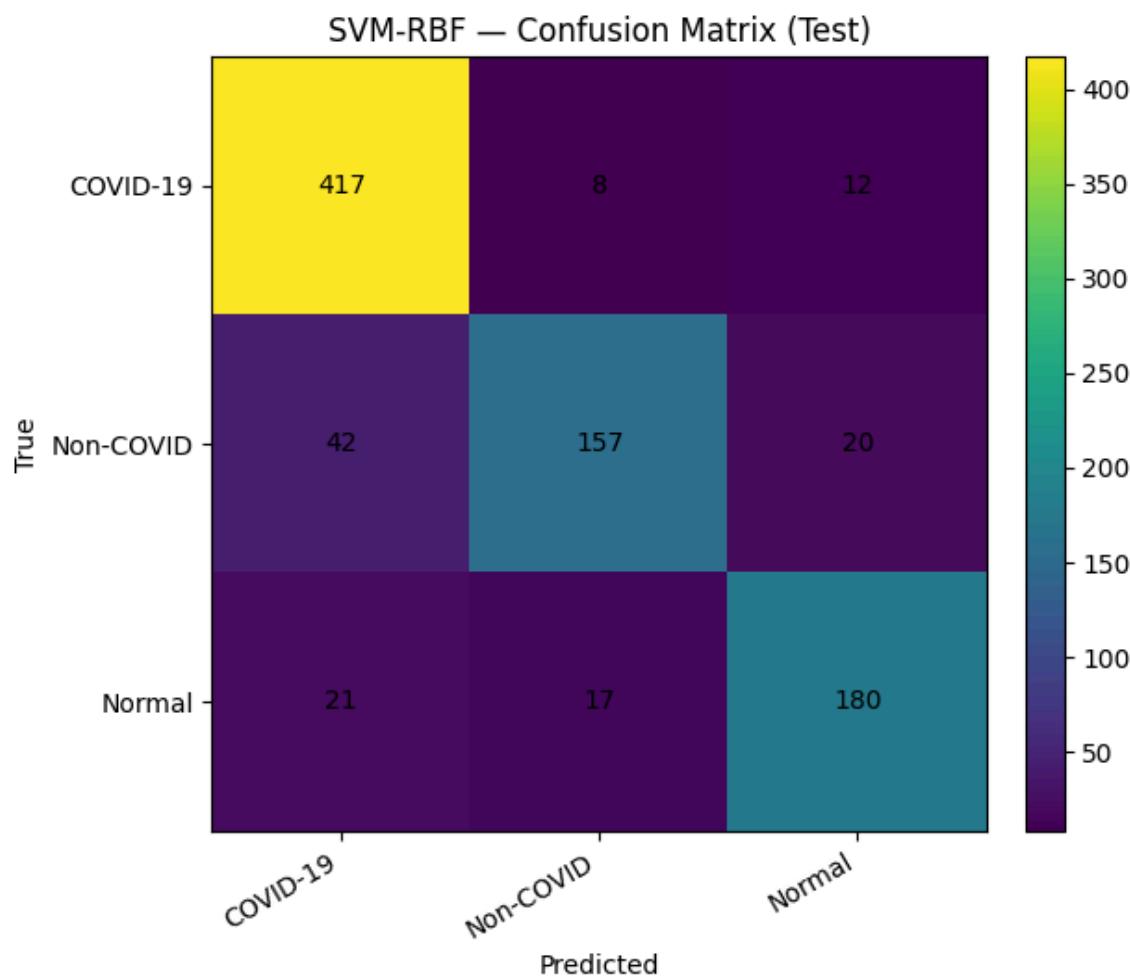
4.1 Tujuan dan Ruang Lingkup

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.

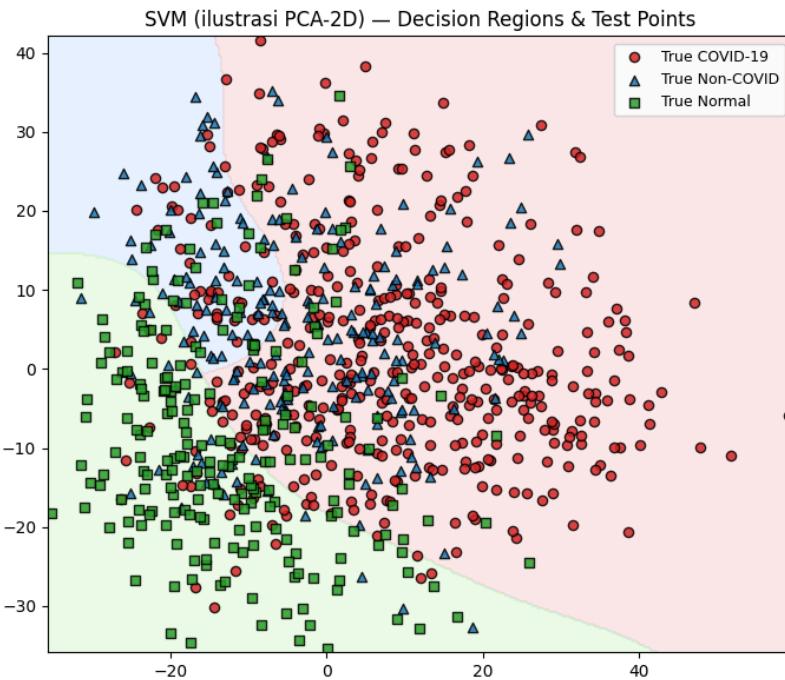
4.2 Analisis Model SVM-RBF

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.



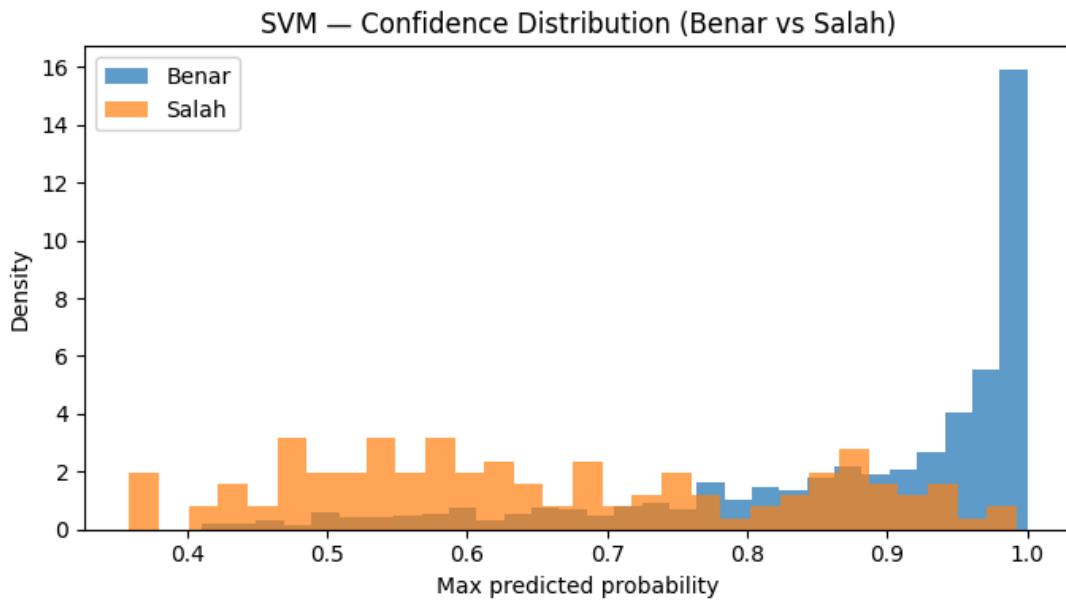
Gbr. 4.1 Confusion Matrix SVM-RBF (Test)

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.



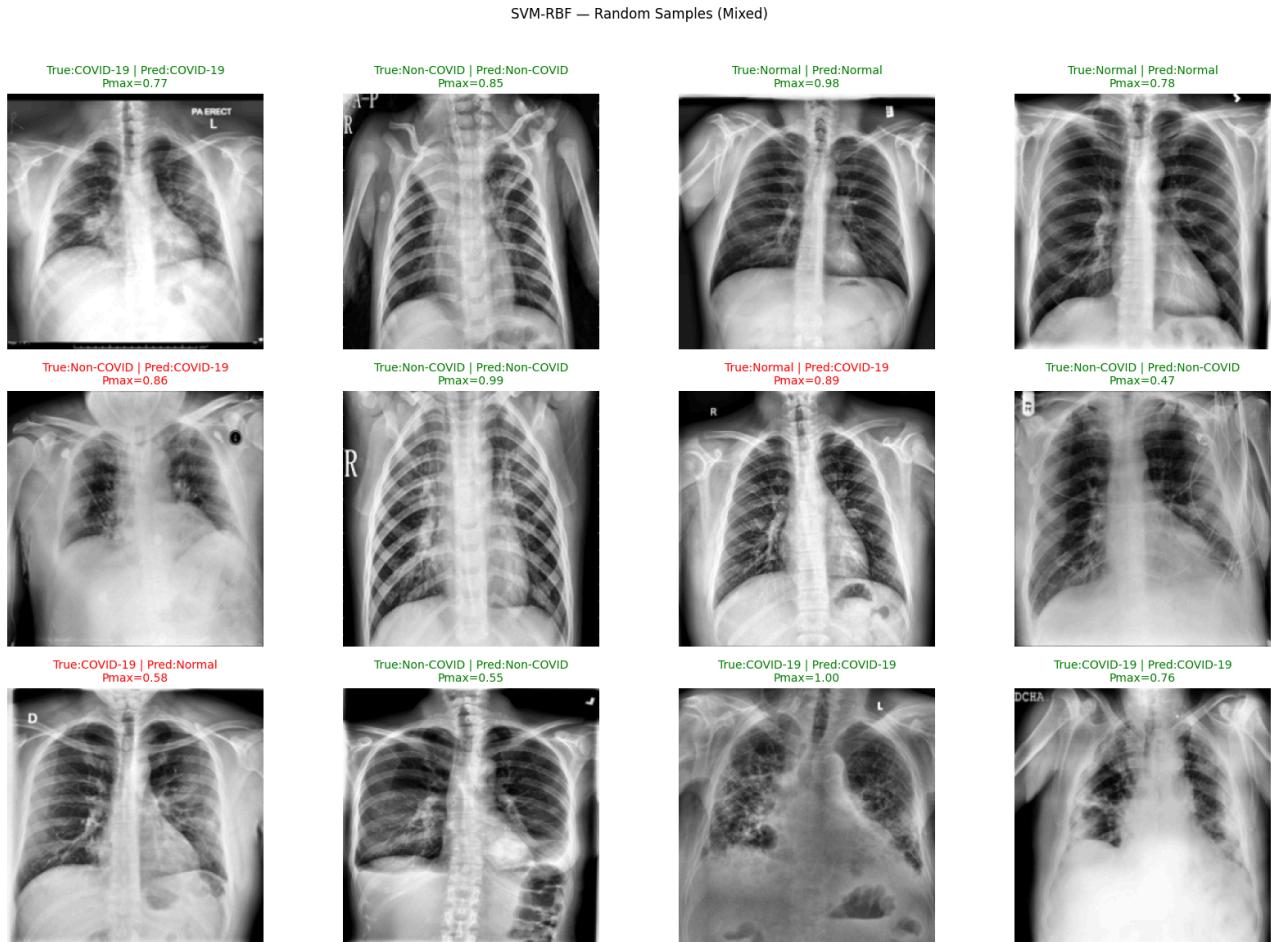
Gbr. 4.2 SVM (ilustrasi PCA-2D) — Decision Regions & Test Points

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.

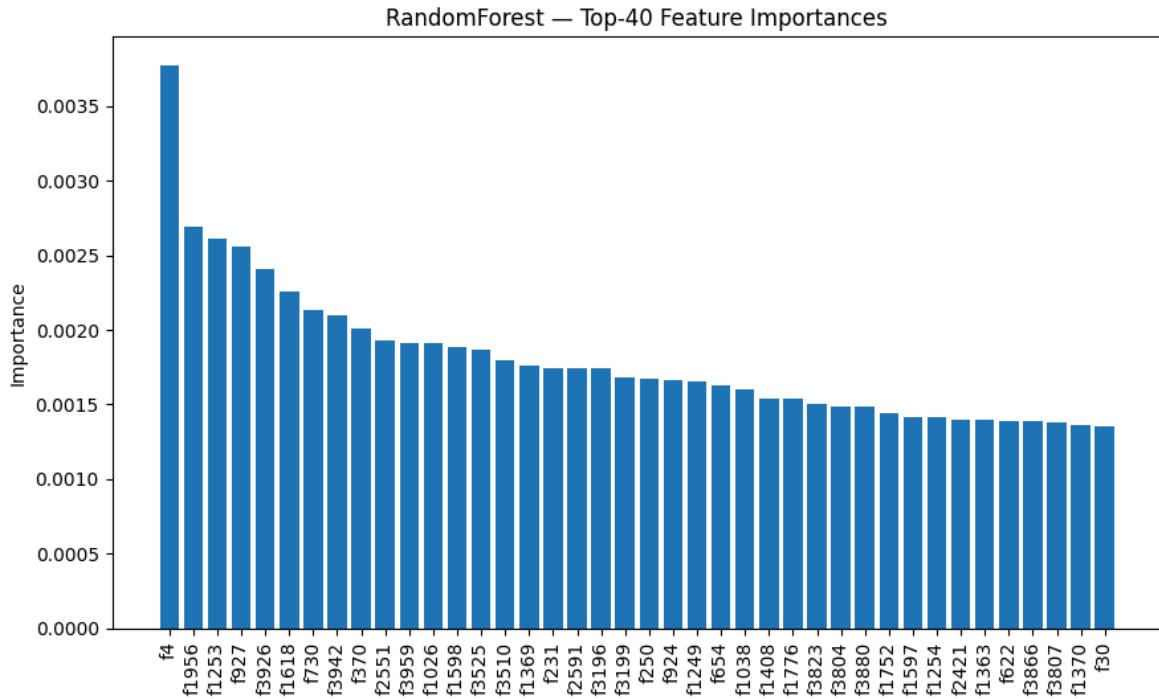


Gbr. 4.4 Sampel Prediksi SVM RBF — Benar vs Salah dengan Confidence

4.3 Analisis Model Random Forest

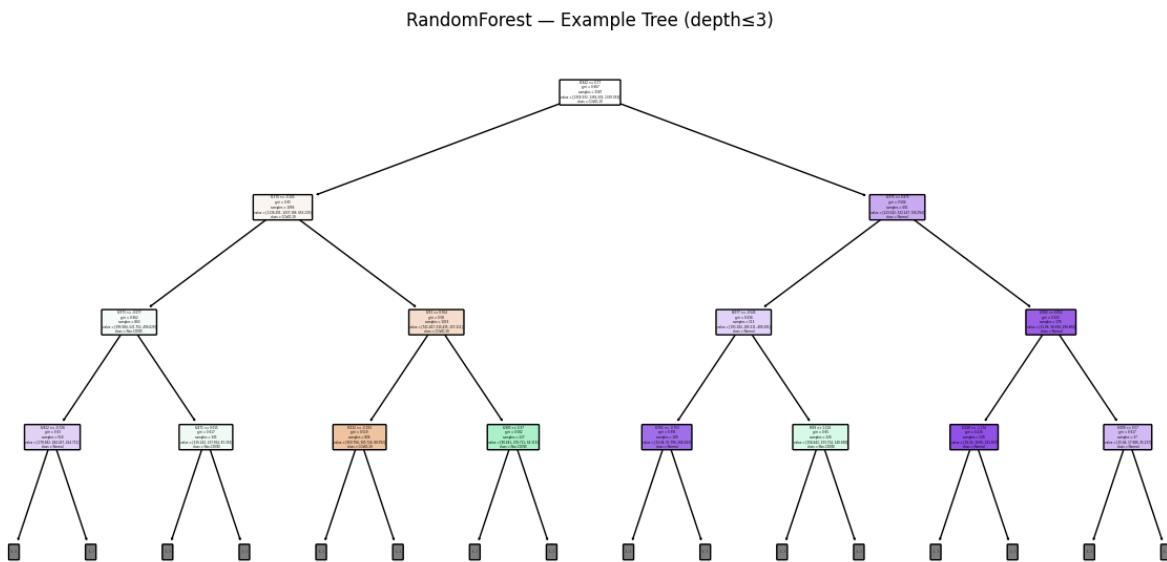
Random Forest mencapai akurasi 76 % | macro-F1 0,719. Model cenderung stabil tetapi tidak sekuat SVM karena keterbatasan dalam menangkap hubungan non-linier antar-fitur.

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.



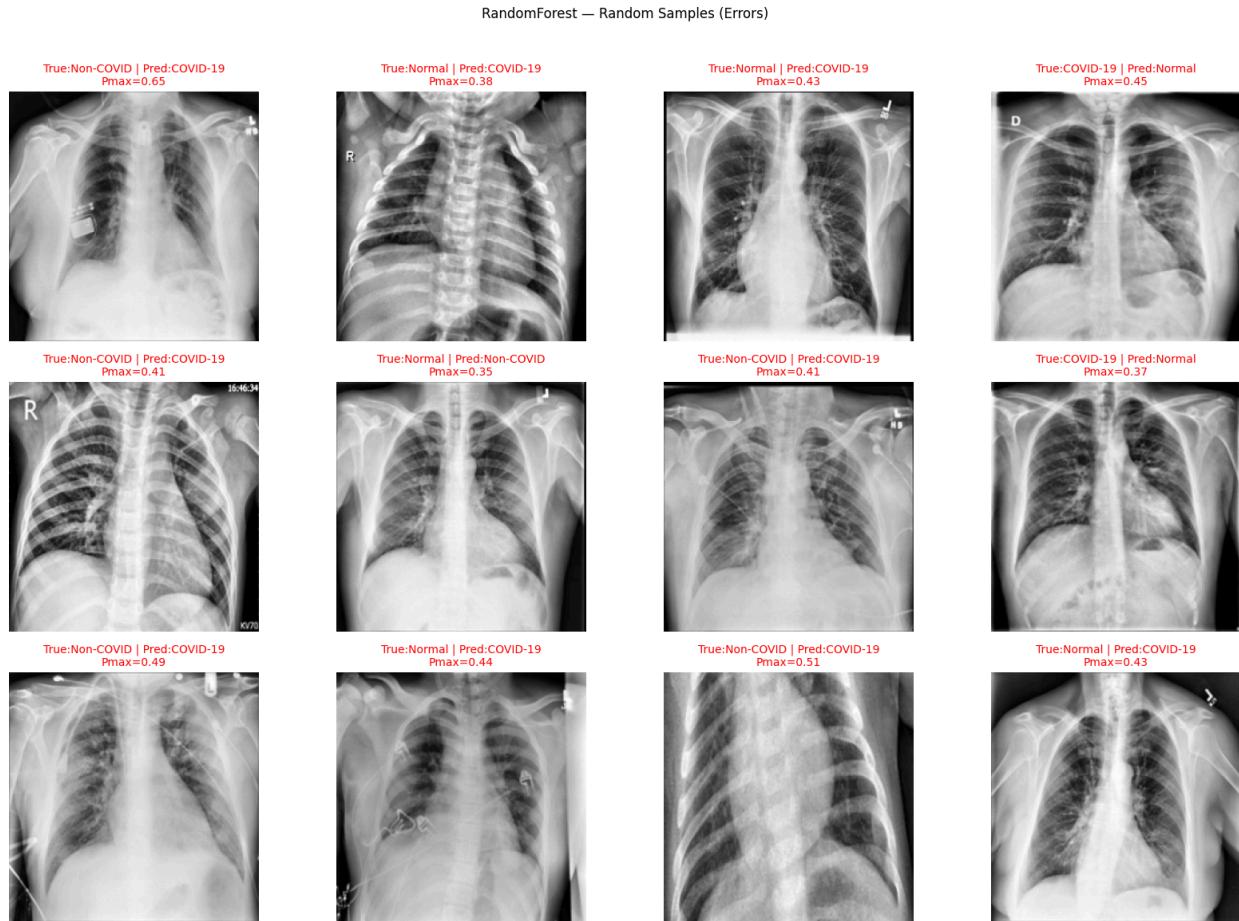
Gbr. 4.5 RandomForest — Top-40 Feature Importances

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.

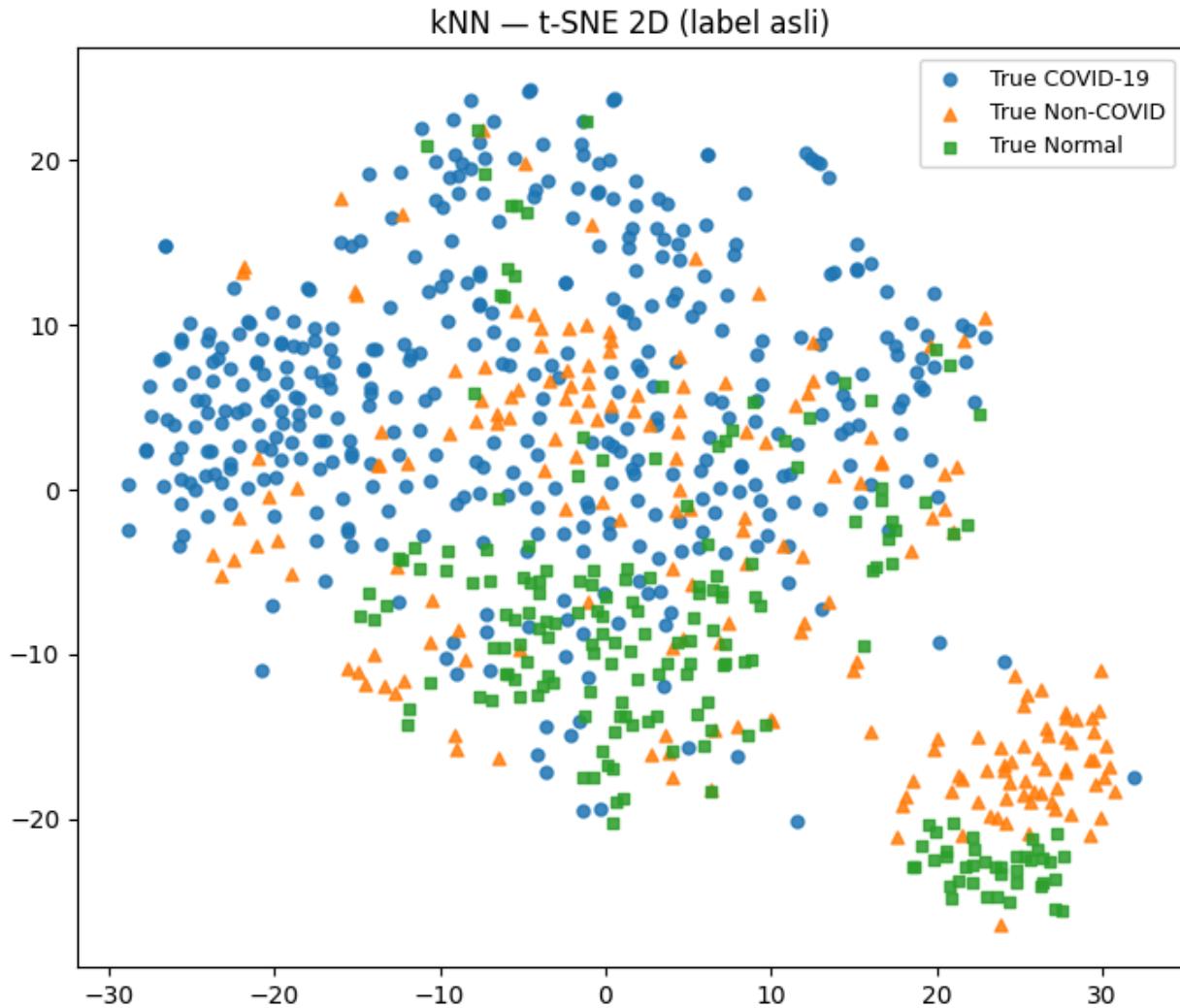


Gbr. 4.7 Sampel Prediksi RandomForest (Test)

4.4 Analisis Model k-Nearest Neighbors (kNN)

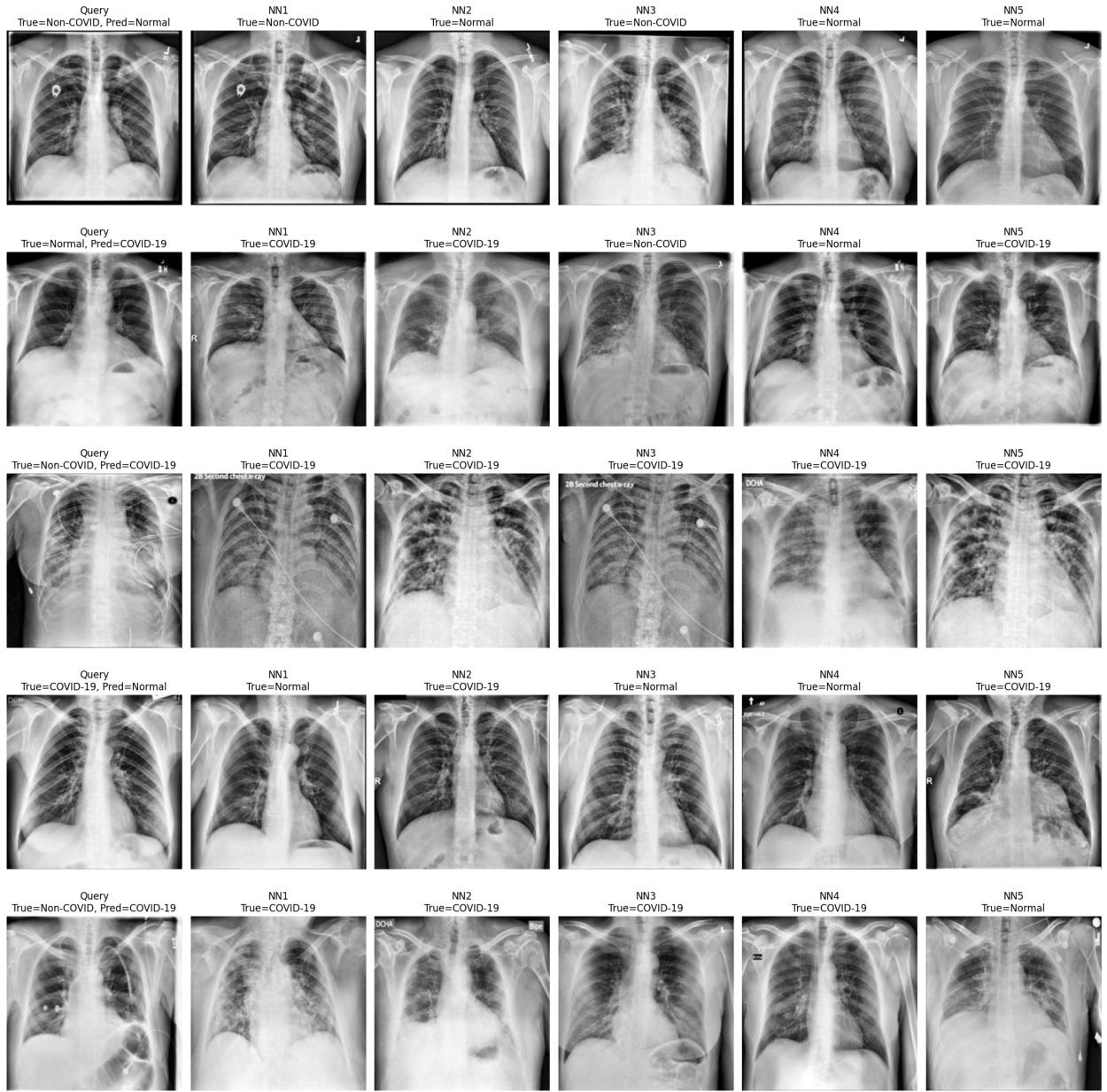
Model kNN dengan $k = 5$ memiliki akurasi 77 % | macro-F1 0,739. Model ini sangat bergantung pada jarak fitur sehingga lebih rentan terhadap distribusi fitur yang padat.

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)

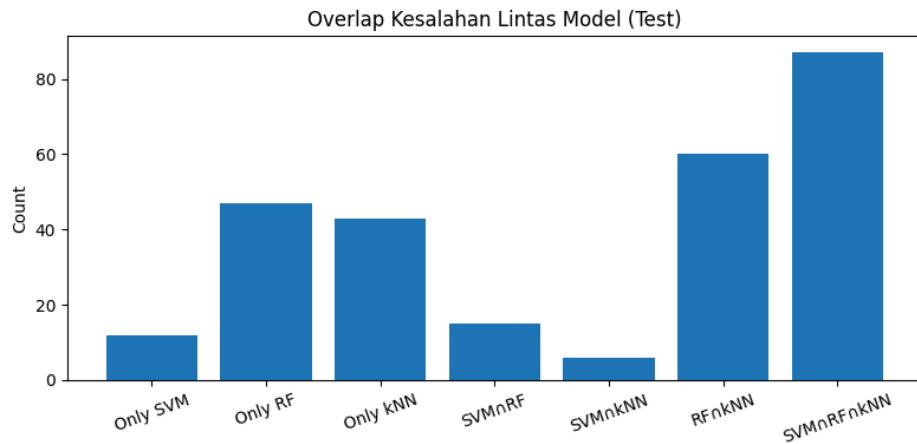
Analisis **5 tetangga terdekat** pada kasus salah (Gbr. 4.9) mengungkap bahwa beberapa citra salah diklasifikasi karena tetangga terdekat berasal dari kelas yang salah tetapi memiliki tekstur paru serupa — misal, COVID-19 dengan Non-COVID tanpa infiltrasi.



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

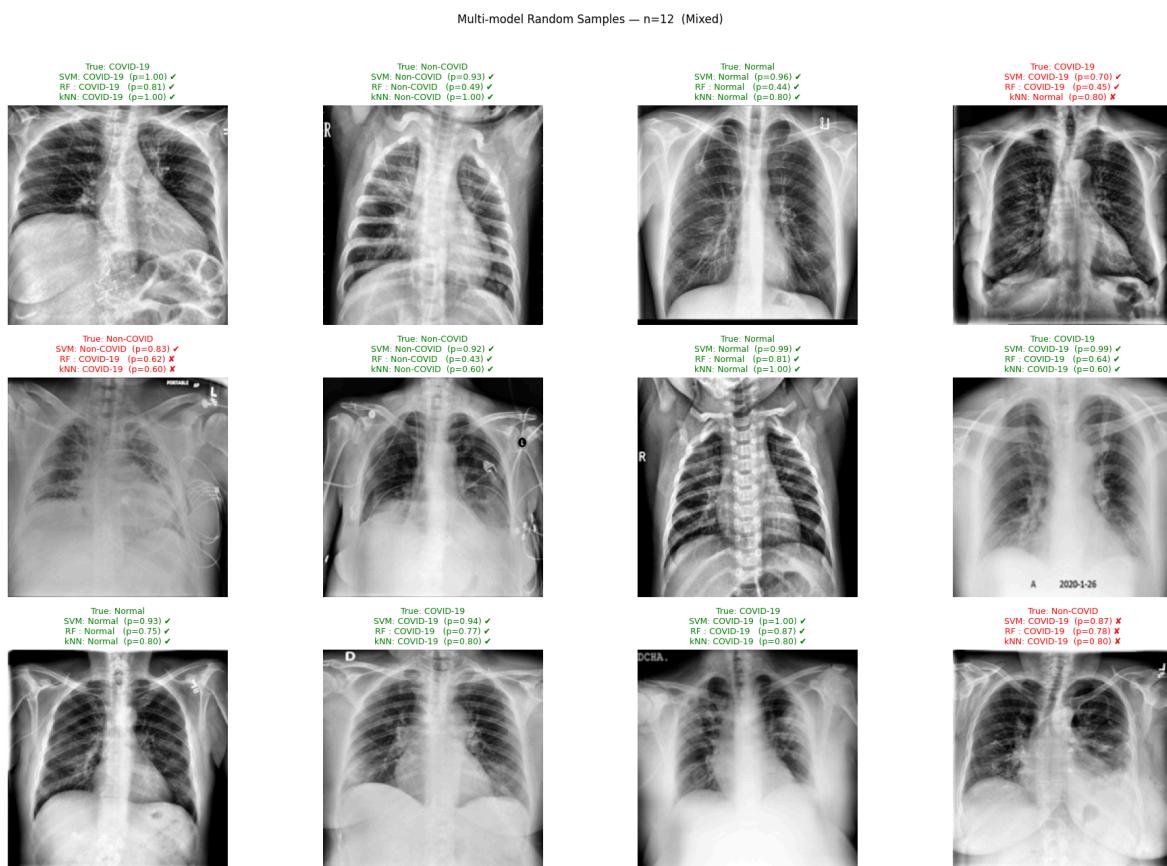
4.5 Analisis Lintas Model

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.



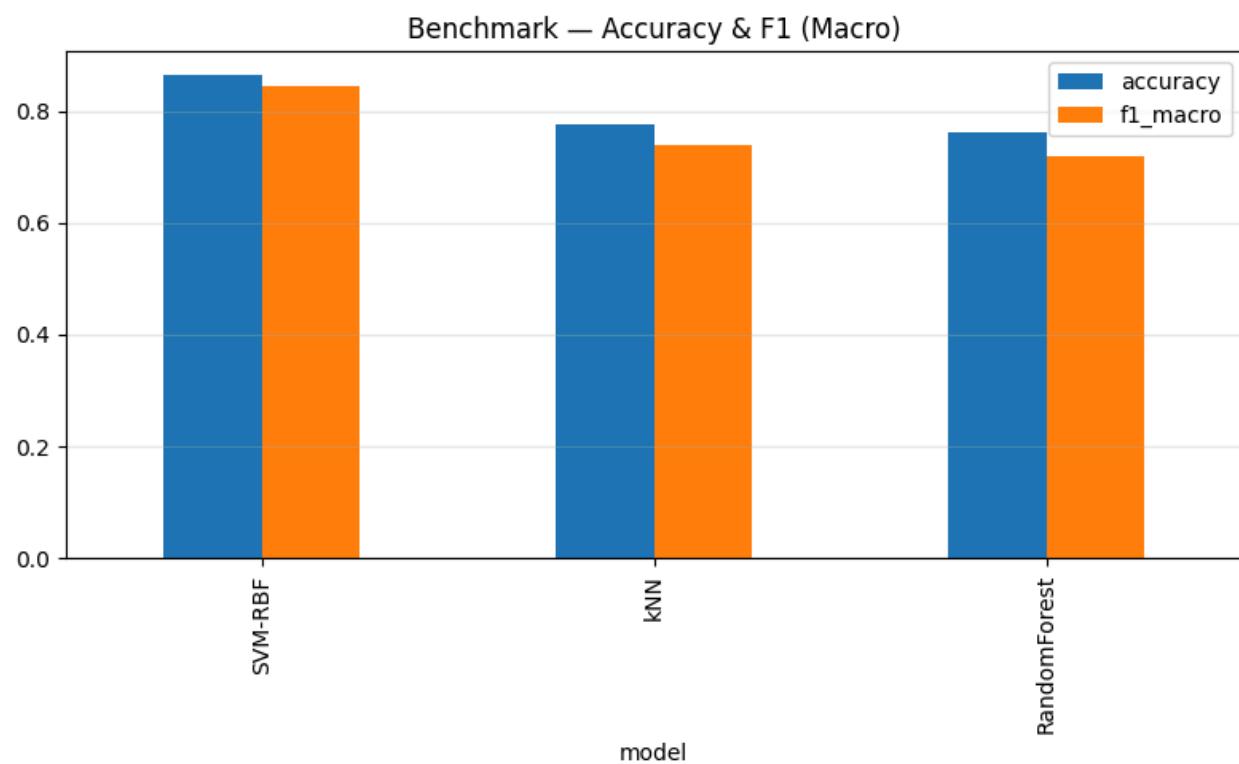
Gbr. 4.11 Grid Multi-Model Random Samples (Benar vs Salah)

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

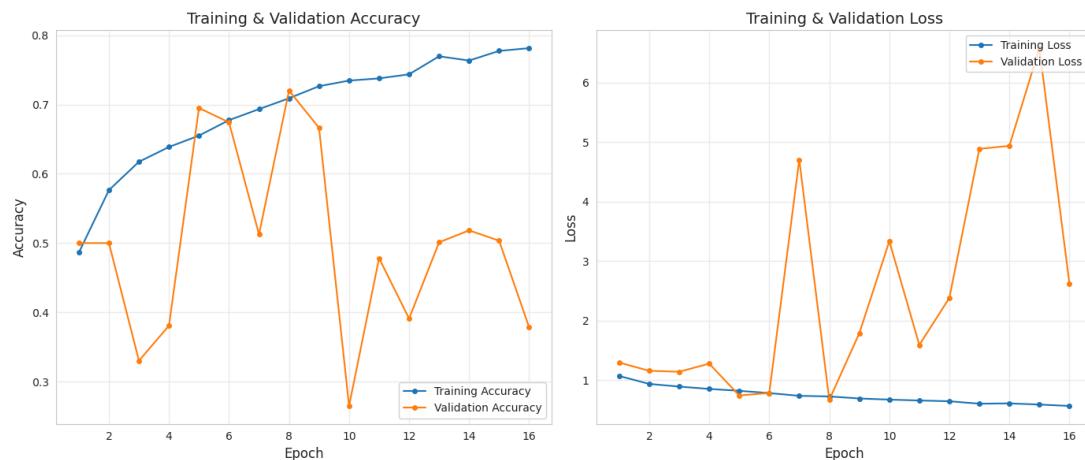
Model dilatih menggunakan data yang sama dengan tahap sebelumnya (Stratified Split). Mengingat ketidakseimbangan kelas (COVID-19 mendominasi 50% data), kami menerapkan Class Weights secara ketat pada fungsi *loss*:

- 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.



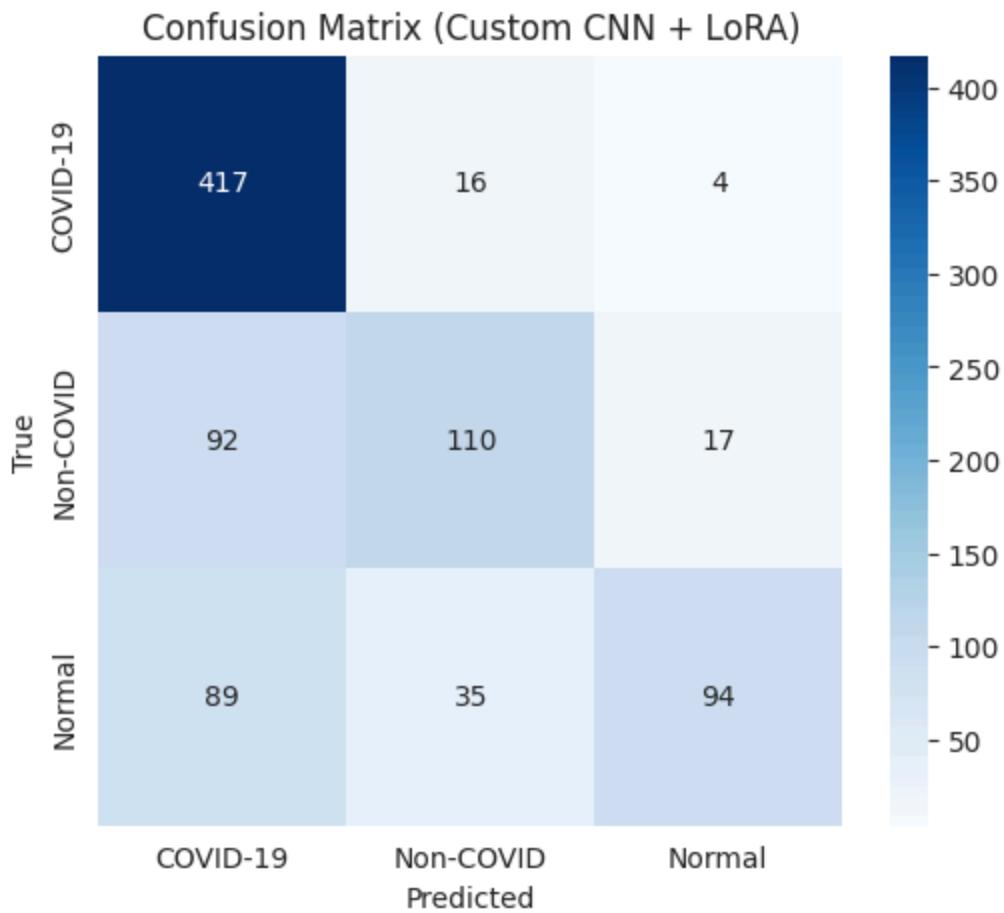
Gbr 5.1. Grafik Akurasi dan Loss selama Pelatihan (Train vs Validation).

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 |
|----------------------|-----------|---------------|---------------|
| COVID-19 | 0.6973 | 0.9542 | 0.8058 |
| Non-COVID | 0.6832 | 0.5023 | 0.5789 |
| Normal | 0.8174 | 0.4312 | 0.5646 |
| Akurasi Total | | | 71.05% |



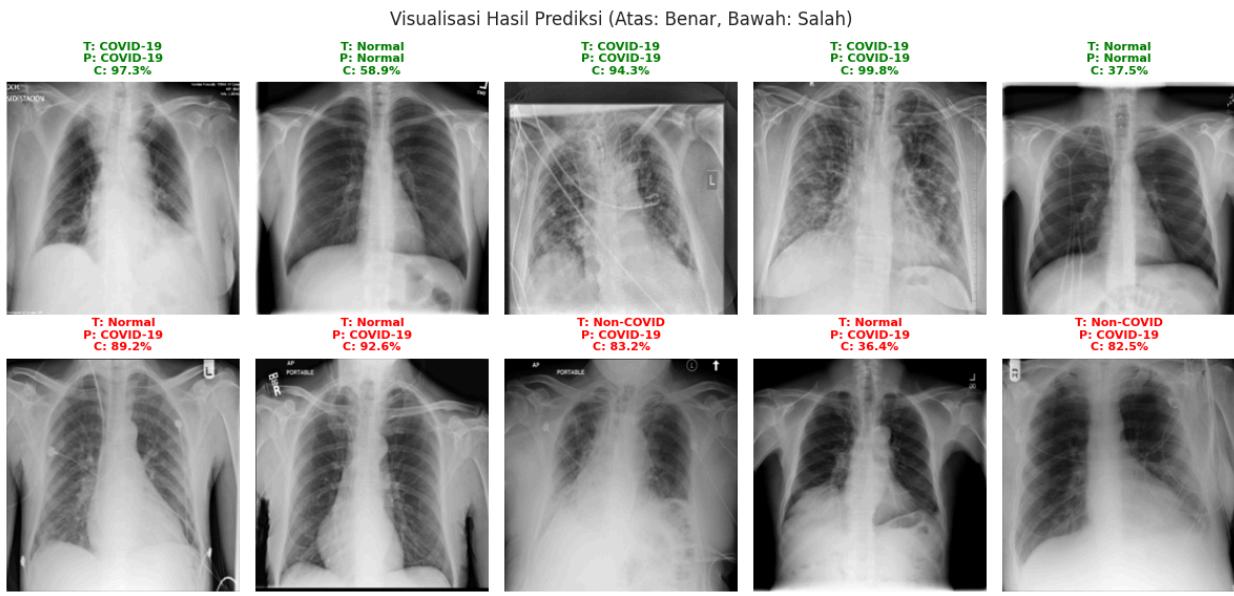
Gbr 5.2. Confusion Matrix pada Data Uji.

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.