

Machine Learning A

Anggota: 202210370311123 – Maylani Kusuma Wardhani

202210370311158 – Garin Muhammad Akbar

1. Business Objective

Tujuan dari penelitian ini adalah untuk mengembangkan sistem berbasis deep learning yang mampu mengenali pola visual pada citra wajah anak-anak sebagai indikator awal status pertumbuhan secara non-medis.

Dengan menerapkan konsep Visual Proxy (VP), sistem ini hanya mempelajari pola proporsi wajah seperti perbedaan jarak antar fitur wajah sebagai representasi visual dari kategori pertumbuhan, yaitu:

- VP-0 (Visually Proportional) → proporsi wajah normal
- VP-1 (Visually Linear) → indikasi perbedaan proporsi wajah

Implementasi ini diharapkan dapat meningkatkan efisiensi deteksi dini potensi gangguan pertumbuhan anak melalui analisis visual sederhana, mendukung pengambilan keputusan berbasis data, serta menjaga etika dan privasi medis karena klasifikasi dilakukan berdasarkan pola visual, bukan diagnosis gizi medis.

2. Urgensi Topik

Gangguan pertumbuhan pada anak, seperti stunting, masih menjadi permasalahan serius yang berdampak pada kualitas hidup dan perkembangan jangka panjang. Deteksi dini menjadi hal penting, namun metode konvensional masih bergantung pada data medis dan tenaga ahli yang memerlukan waktu serta sumber daya besar. Oleh karena itu, penelitian ini penting karena menawarkan pendekatan deep learning berbasis Visual Proxy (VP) untuk mengenali pola visual wajah anak secara cepat dan non-medis, sehingga dapat mempercepat proses pemantauan pertumbuhan sekaligus menjaga etika serta privasi data anak.

3. Dataset

Dataset diambil dari platform **Roboflow**, yang berisi citra wajah anak-anak dengan dua kategori: *Healthy* dan *Stunting*. Dataset ini telah dikonversi dan diorganisasi ulang menggunakan **label proxy (VP-0 dan VP-1)**. Link [Dataset](#)



Gambar 1. Sample data sebelum dilakukan preprocessing

4. Data Preprocessing

Tahap preprocessing dilakukan untuk menyiapkan data sebelum masuk ke model deep learning. Langkah-langkahnya meliputi:

4.1. Loading Dataset

Dataset dimuat dari GitHub repository menggunakan gdown atau requests agar dapat digunakan langsung di Google Colab.

4.2. Ekstraksi Wajah

Setiap citra diproses untuk mendeteksi dan memotong area wajah menggunakan detektor wajah **MTCNN**. Tujuan: memastikan model hanya belajar dari area wajah yang relevan.

4.3. Resize dan Normalisasi

- Gambar diubah ukurannya menjadi dimensi standar (224x224 px).
- Nilai piksel dinormalisasi ke rentang [0, 1] atau [-1, 1] untuk mempercepat konvergensi model.

4.4. Data Augmentation

Untuk memperkaya variasi visual dan mengurangi overfitting, diterapkan augmentasi seperti:

- Random rotation (rotasi acak)

- Horizontal flip (membalik gambar)
- Brightness & contrast adjustment
- Random crop (potongan acak)

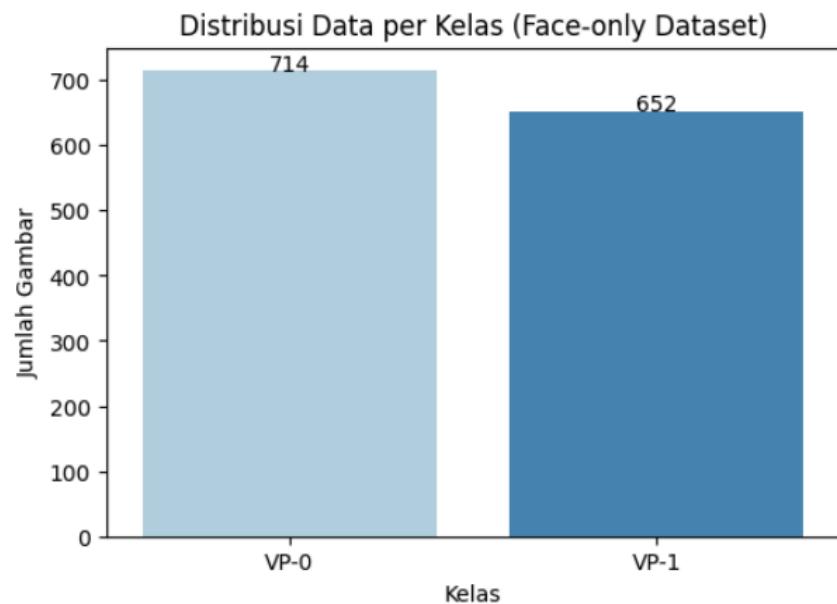
4.5. Pembagian Data

Dataset dibagi menjadi:

- **Training set:** 80%
- **Test set:** 20%

Pembagian dilakukan secara **stratified** agar distribusi VP-0 dan VP-1 tetap seimbang.

4.6. Visualisasi



Gambar 2. Visualisasi distribusi data per kelas



Gambar 3. Visualisasi Tahap Augmentasi Data



Gambar 4. Visualisasi Sample Data setelah Preprocessing (VP-0 = Label: 0, VP-1 = Label: 1)



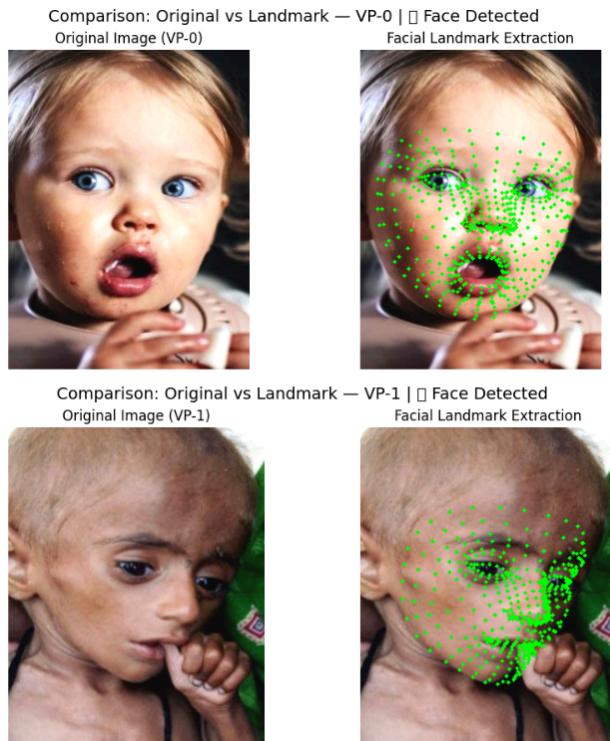
Gambar 5. Perbandingan Sample Data Sebelum dan Sesudah Pre-Processing

5. Feature Extraction

Fitur ekstraksi merupakan tahap penting dalam pengolahan citra yang berfungsi untuk mengambil informasi paling relevan dari gambar agar dapat diolah oleh model klasifikasi. Pada penelitian ini, fitur ekstraksi digunakan untuk mengubah citra wajah menjadi representasi numerik yang menggambarkan bentuk, tekstur, dan warna kulit, sehingga model dapat membedakan antara wajah anak dengan kondisi VP-0 dan VP-1 secara lebih akurat. Proses ini dilakukan menggunakan beberapa metode, yaitu Landmark untuk menangkap struktur dan proporsi wajah, GLCM untuk menganalisis tekstur kulit dengan tambahan Gaussian Blur dan CLAHE agar kontras lebih stabil, CCM untuk melihat distribusi warna kulit dengan bantuan Bilateral Filter dan CLAHE, serta SIFT untuk mendeteksi pola tepi dan titik khas wajah.

5.1. Landmark Extraction

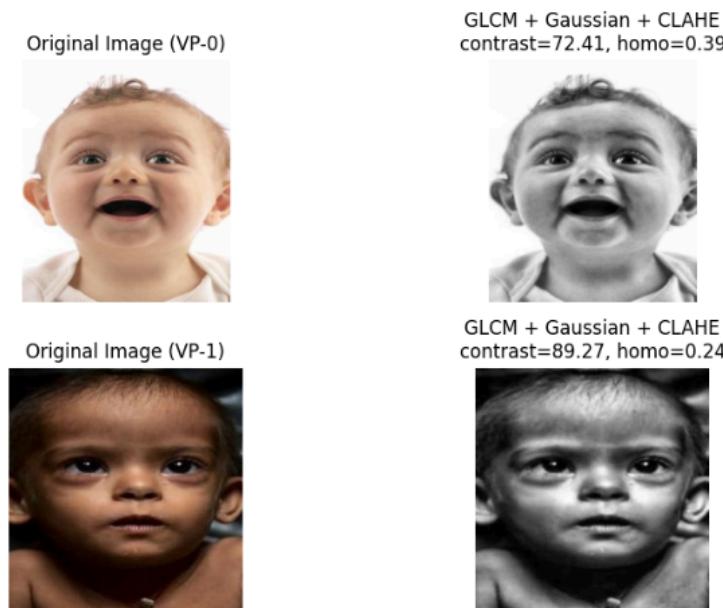
Ekstraksi landmark bertujuan menangkap titik-titik kunci wajah (mis. sudut mata, ujung hidung, sudut mulut, tepi rahang) yang merepresentasikan bentuk dan proporsi anatomi wajah. Titik-titik ini diperoleh dengan detektor landmark yang di-fit ke wajah ter-deteksi (face detection), kemudian diproses menjadi vektor posisi relatif (x,y) atau fitur turunan seperti jarak dan rasio antar-titik. Fitur landmark sangat berguna untuk membedakan variasi morfologi wajah yang stabil terhadap perubahan pencahayaan dan ekspresi ringan, sehingga membantu model menangkap informasi geometris yang relevan untuk klasifikasi.



Gambar 6. Visualisasi Sample Data Menggunakan Landmark Extraction

5.2. GLCM + Filter Klasik (Gaussian Blur dan CLAHE)

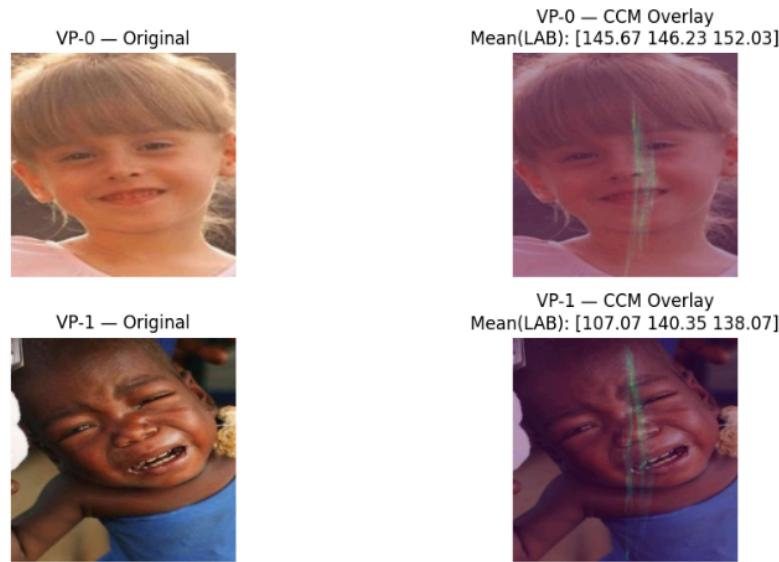
GLCM (Gray-Level Co-occurrence Matrix) digunakan untuk mengekstrak sifat tekstur kulit dengan menghitung hubungan kemunculan pasangan intensitas piksel pada berbagai arah. Sebelum menghitung GLCM, citra diolah dengan Gaussian Blur ringan untuk mereduksi noise halus dan kemudian diterapkan CLAHE (Contrast Limited Adaptive Histogram Equalization) untuk menyeimbangkan kontras lokal, urutan ini membuat pola tekstur lebih stabil dan kontras lebih konsisten antar sampel. Kombinasi GLCM dan pra-pemrosesan ini menghasilkan fitur tekstur (mis. contrast, homogeneity, energy, correlation) yang sensitif terhadap perubahan tekstur kulit namun relatif tahan terhadap variasi pencahayaan.



Gambar 7. Visualisasi Sample Data Menggunakan GLCM + Filter Klasik

5.3. CCM + Filter Klasik (Bilateral Filter dan CLAHE)

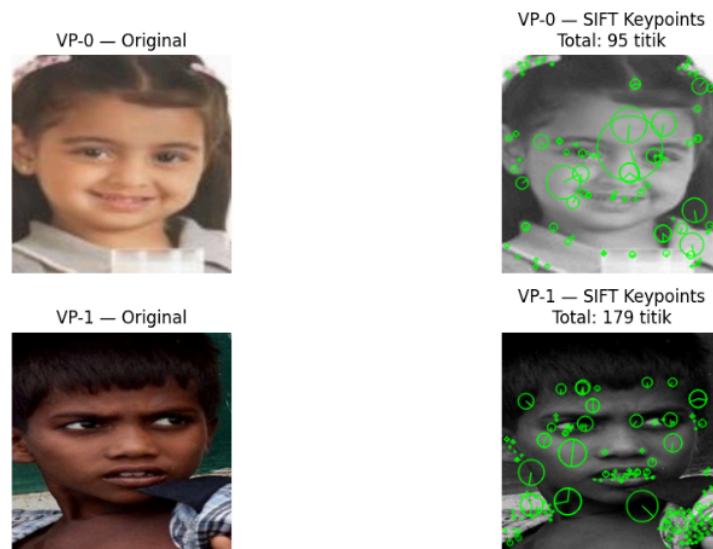
CCM (Color Co-occurrence Matrix) memodelkan distribusi warna dan hubungan antar kanal warna, biasanya di ruang warna LAB untuk memisahkan informasi luminansi dan krominans. Sebelum ekstraksi, citra diberi bilateral filter untuk menghaluskan warna kulit tanpa mengaburkan tepi penting, lalu channel L diperkaya dengan CLAHE untuk menstabilkan pencahayaan lokal. Pendekatan ini menjaga detail warna fisiologis (mis. rona pipi, perbedaan bayangan) dan menghasilkan fitur warna statistik yang membantu membedakan kondisi kulit yang mungkin berkorelasi dengan label diagnostik.



Gambar 8. Visualisasi Sample Data Menggunakan CCM + Filter Klasik

5.4. SIFT

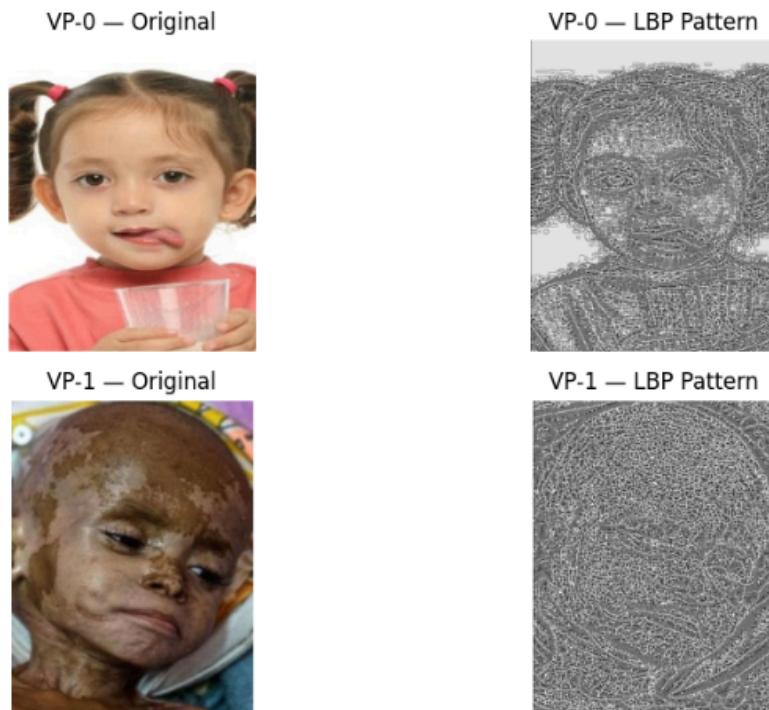
SIFT mengekstraksi fitur titik kunci lokal yang tahan terhadap skala dan rotasi dengan mendeteksi corner/sudut dan mendeskripsikannya menggunakan deskriptor berukuran tetap (biasanya 128 dimensi). Untuk wajah, SIFT menangkap pola tepi, lipatan, dan tekstur mikro yang tidak selalu ditangkap oleh statistik global, fitur ini berguna untuk mengenali detail struktural kecil yang khas pada sub-area wajah. Karena SIFT sensitif terhadap over-smoothing, pra-pemrosesan untuk SIFT sebaiknya minimal (atau menggunakan sharpening ringan) sehingga titik kunci tetap terdeteksi dengan andal.



Gambar 9. Visualisasi Sample Data Menggunakan SIFT

5.5. LBP

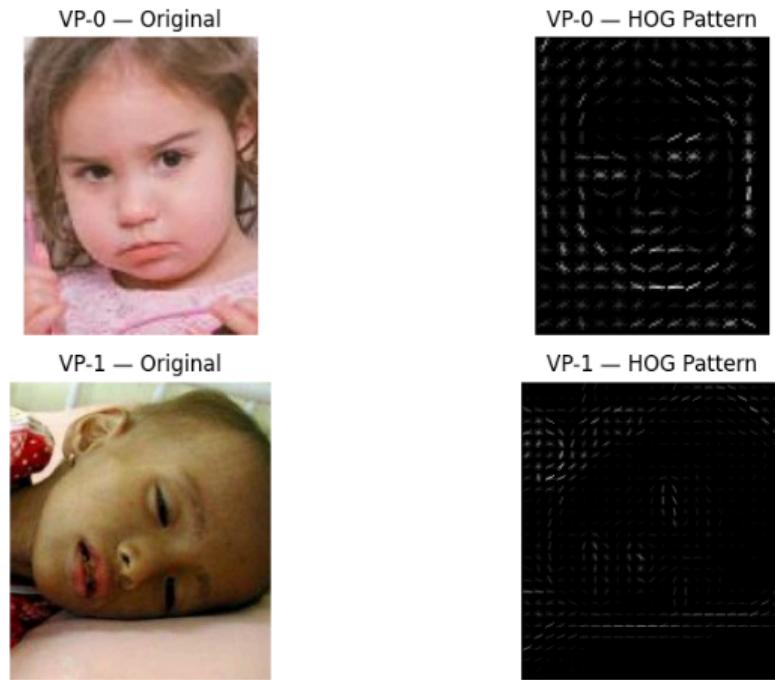
LBP mengekstraksi fitur tekstur lokal dengan membandingkan intensitas piksel pusat terhadap piksel di sekitarnya dalam pola melingkar. Setiap piksel diubah menjadi representasi biner berdasarkan apakah intensitas tetangganya lebih besar atau lebih kecil dari piksel pusat, menghasilkan kode biner yang menggambarkan mikro-pola tekstur seperti tepi, sudut, atau area datar.



Gambar 10. Visualisasi Sample Data Menggunakan LBP

5.6. HOG

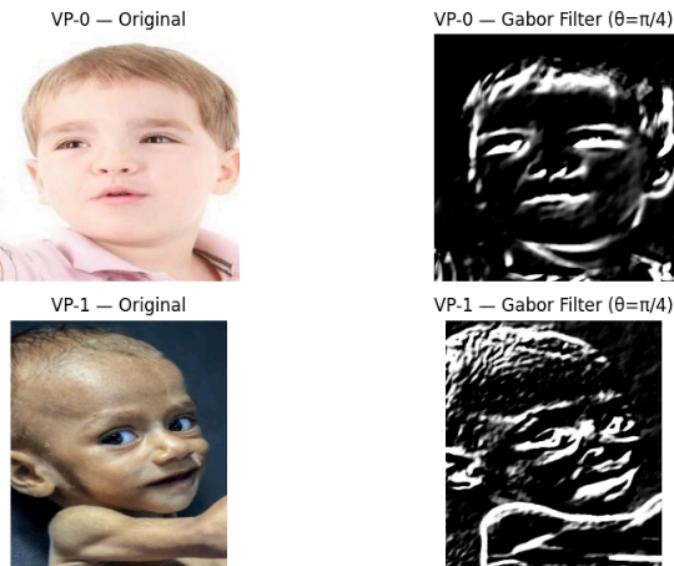
HOG mengekstraksi fitur dengan menghitung distribusi orientasi gradien pada sel-sel kecil dalam citra. Metode ini memecah gambar menjadi blok dan menganalisis arah dominan tepi di setiap wilayah. Hasilnya adalah representasi struktur bentuk (shape) yang kuat terhadap perubahan pencahayaan dan kecil terhadap rotasi.



Gambar 11. Visualisasi Sample Data Menggunakan HOG

5.7. Gabor

Gabor filter mengekstraksi informasi tekstur dan frekuensi spasial pada berbagai skala dan orientasi dengan menggunakan fungsi sinusoidal termodulasi Gaussian. Filter ini meniru mekanisme persepsi visual manusia dalam mendeteksi tepi dan arah pola tekstur.



Gambar 12. Visualisasi Sample Data Menggunakan Gabor

5.8. Color Moment

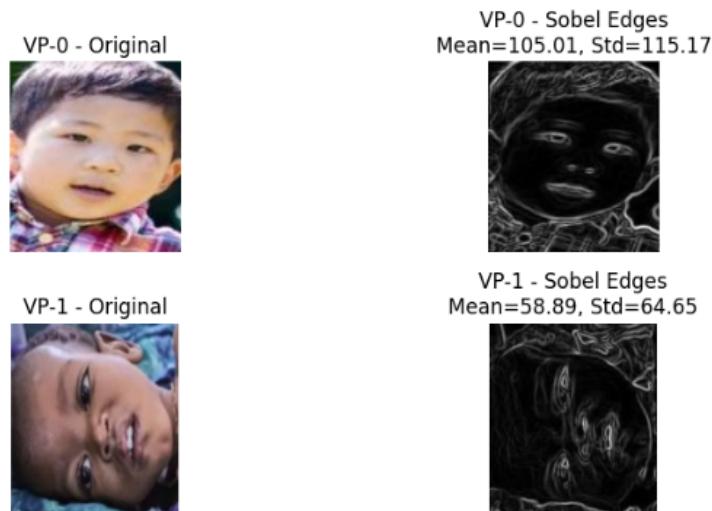
Color Moment mengekstraksi informasi warna dengan menghitung momen statistik (mean, standard deviation, dan skewness) dari distribusi warna pada setiap kanal (misalnya RGB atau HSV). Momen ini merepresentasikan karakteristik global warna dalam suatu gambar secara kompak.



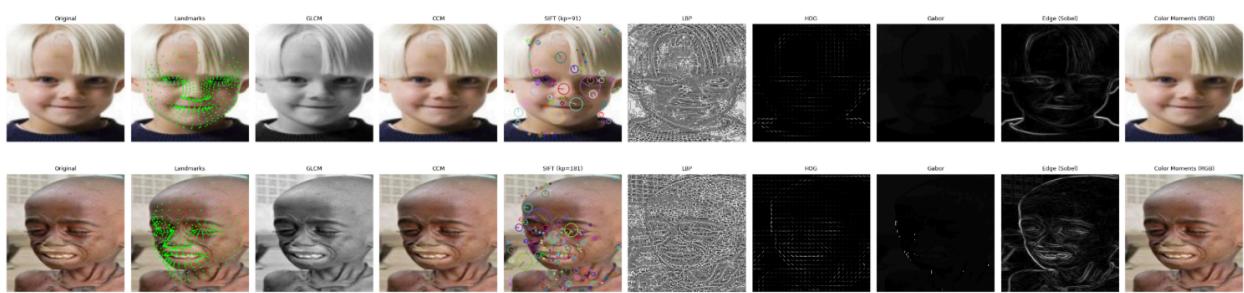
Gambar 13. Visualisasi Sample Data Menggunakan Color Moment

5.9. Edge (Sobel)

Operator Sobel digunakan untuk mendeteksi tepi dengan menghitung gradien intensitas piksel secara horizontal dan vertikal. Hasilnya menggambarkan area transisi intensitas tinggi yang biasanya menunjukkan batas objek atau kontur wajah.



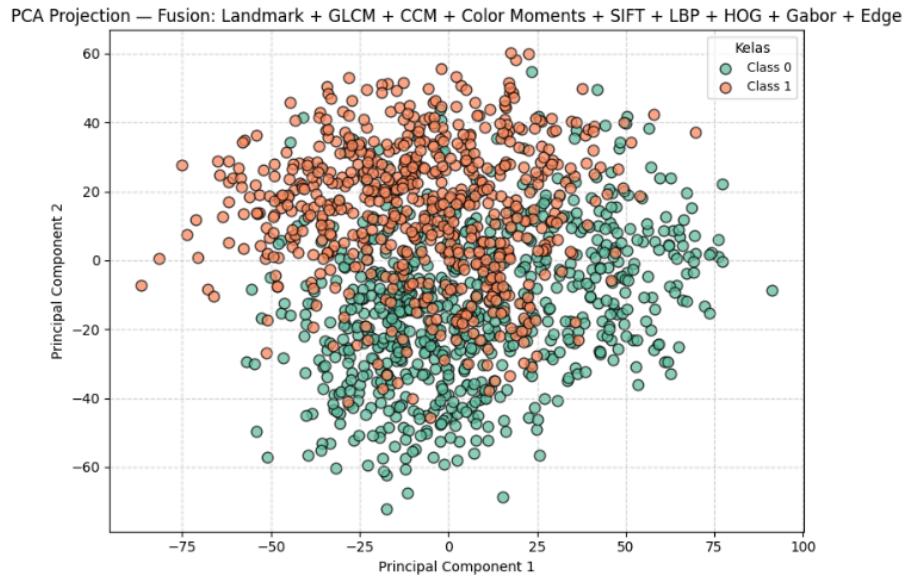
Gambar 14. Visualisasi Sample Data Menggunakan Edge



Gambar 15. Visualisasi Keseluruhan Tahap Fitur Extraction

5.10. Feature Fusion

Feature Fusion dilakukan dengan cara menggabungkan seluruh fitur yang telah diekstraksi yaitu landmark, GLCM, CCM, dan SIFT menjadi satu vektor fitur terpadu.



Gambar 16. Visualisasi Proyeksi PCA pada Data Fusion

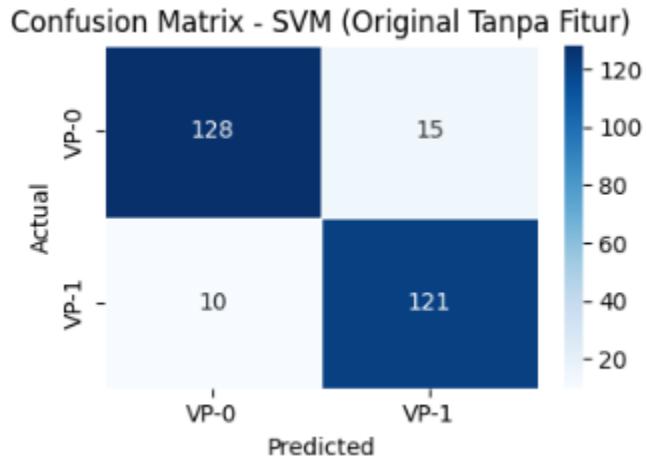
6. Model Training and Evaluation

6.1. Uji Klasifikasi dengan Model Klasik

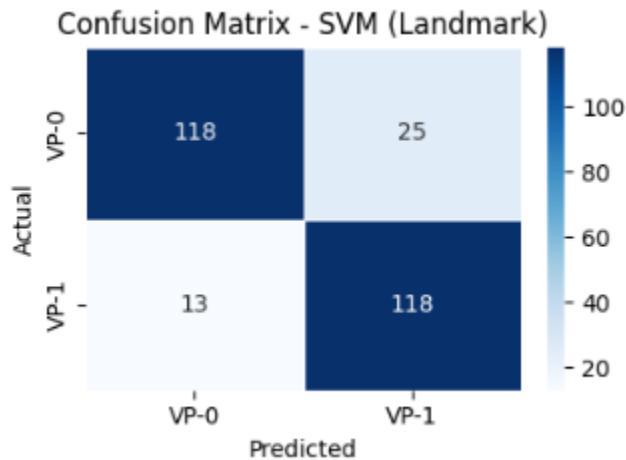
Tiga model klasifikasi klasik digunakan untuk mengevaluasi performa hasil ekstraksi fitur, yaitu Support Vector Machine (SVM), Random Forest (RF), dan K-Nearest Neighbors (KNN). Skema pembagian data dilakukan menggunakan train-test split dengan rasio 80:20, di mana 80% data digunakan untuk pelatihan dan 20% untuk pengujian. Pembagian ini dilakukan secara stratified, artinya proporsi label di data latih dan uji tetap seimbang.

6.1.1. SVM

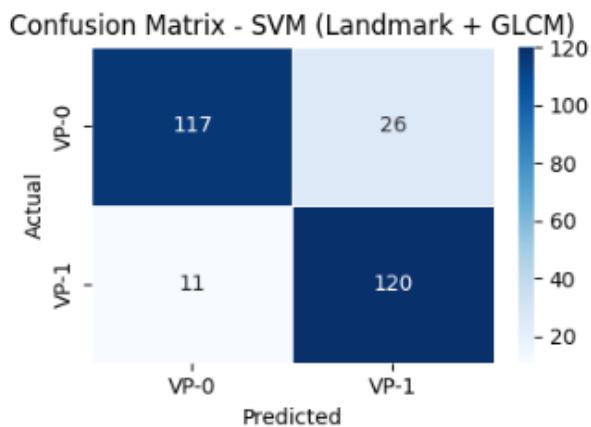
6.1.1.1. SVM-only



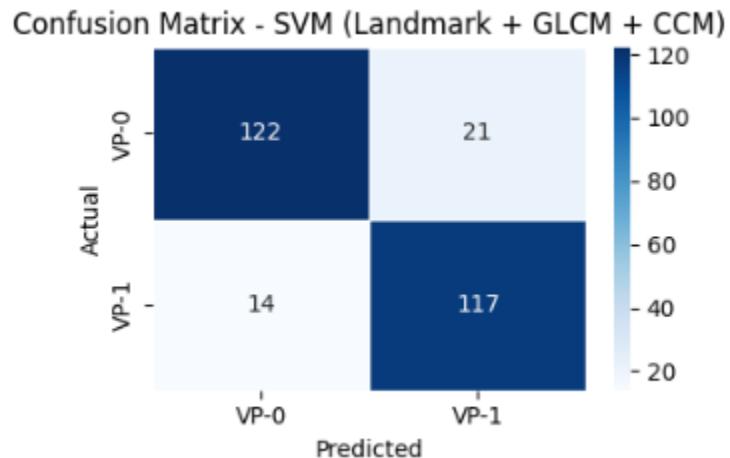
6.1.1.2 SVM+LANDMARK



6.1.1.3 SVM+LANDMARK+GLCM

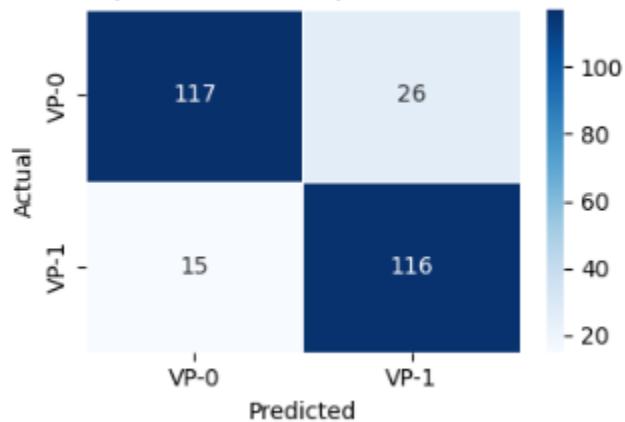


6.1.1.4 SVM+LANDMARK+GLCM+CCM

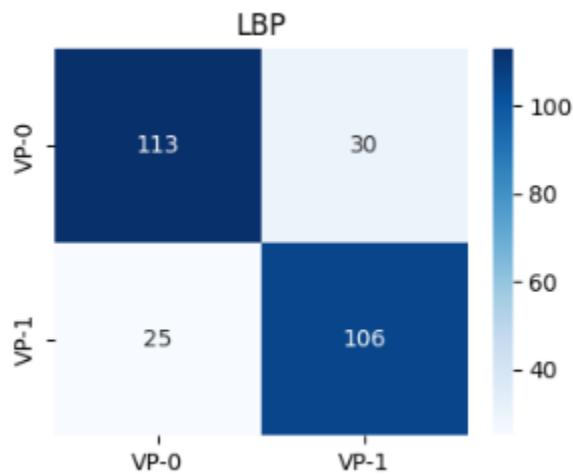


6.1.1.5 SVM+LANDMARK+GLCM+CCM+SIFT

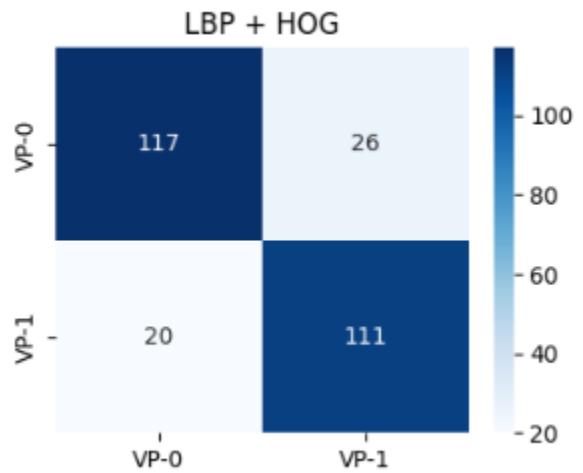
Confusion Matrix - SVM (Feature Fusion (Landmark + GLCM + CCM + SIFT))



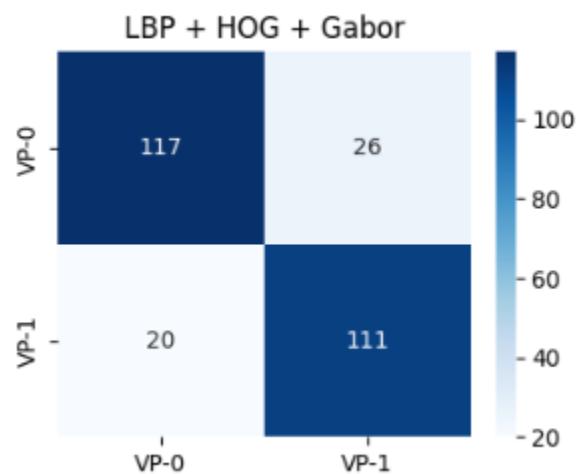
6.1.1.6 SVM+LBP



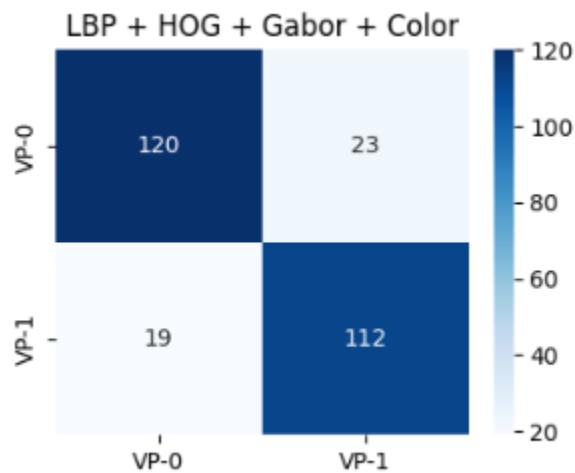
6.1.1.7 SVM+LBP+HOG



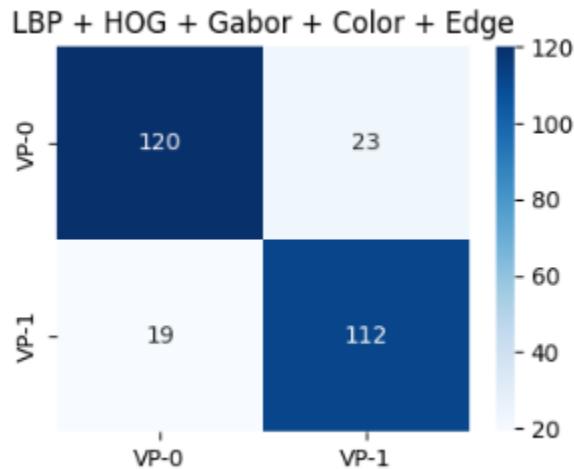
6.1.1.8. SVM+LBP+HOG+GABOR



6.1.1.9. SVM+LBP+HOG+GABOR+COLOR



6.1.1.10. SVM+LBP+HOG+GABOR+COLOR+EDGE



Gambar 17. Visualisasi Confusion Matrix Keseluruhan untuk Model SVM Baseline dan Fusion

Tabel 1. Model Summary SVM

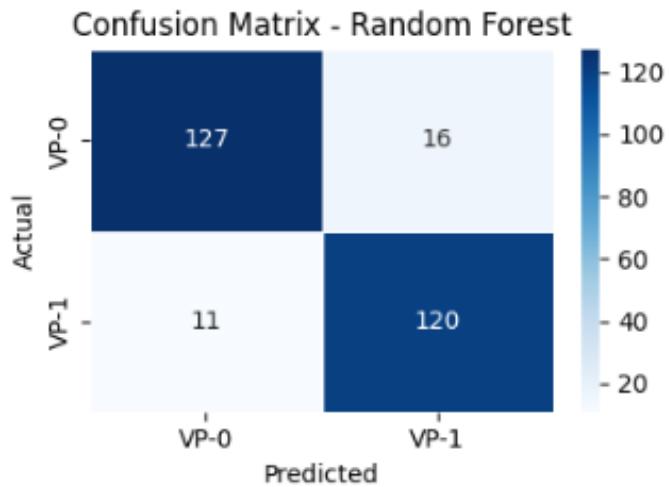
Model Fitur	Algoritma	Akurasi	Precision	Recall	F1-Score
Baseline	SVM	0.91	0.93	0.90	0.91
Landmark	SVM	0.86	0.90	0.83	0.86
Landmark+GLCM	SVM	0.86	0.91	0.82	0.86

Landmark+GLCM+CCM	SVM	0.87	0.90	0.85	0.87
Landmark+GLCM+CCM+SIFT	SVM	0.85	0.89	0.82	0.85
LBP	SVM	0.80	0.82	0.79	0.80
LBP+HOG	SVM	0.83	0.85	0.82	0.84
LBP+HOG+GABOR	SVM	0.83	0.85	0.82	0.84
LBP+HOG+GABOR+COLOR	SVM	0.85	0.86	0.84	0.85
LBP+HOG+GABO R+COLOR+EDGE	SVM	0.85	0.86	0.84	0.85

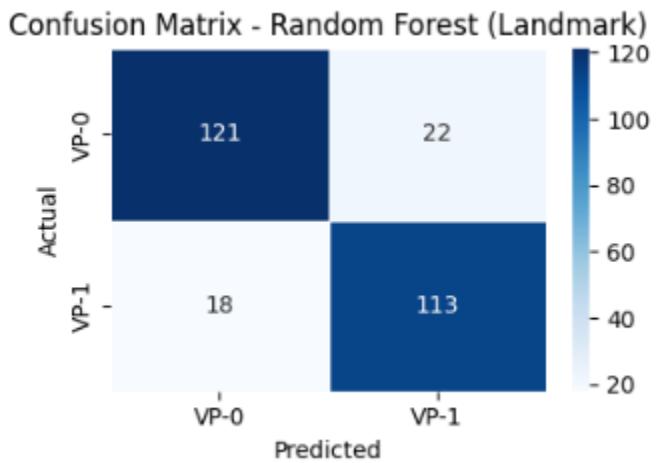
Hasil evaluasi model SVM menunjukkan bahwa model baseline tanpa fitur tambahan memberikan performa terbaik dengan akurasi tertinggi sebesar 0.9088, diikuti oleh model SVM dengan kombinasi Landmark + GLCM + CCM dengan akurasi 0.8723. Penambahan fitur tunggal seperti landmark atau kombinasi sebagian fitur memang meningkatkan stabilitas prediksi antar kelas, tetapi tidak selalu menghasilkan peningkatan akurasi yang signifikan. Berdasarkan confusion matrix, model baseline mampu mengklasifikasikan kedua kelas (VP-0 dan VP-1) dengan seimbang, ditunjukkan oleh nilai precision dan recall yang konsisten sekitar 0.9. Sementara itu, model fusion (Landmark + GLCM + CCM + SIFT) menunjukkan penurunan akurasi menjadi 0.8504, yang mengindikasikan bahwa penambahan fitur SIFT justru menambah kompleksitas tanpa memberikan kontribusi signifikan terhadap kinerja klasifikasi.

6.1.2 Random Forest

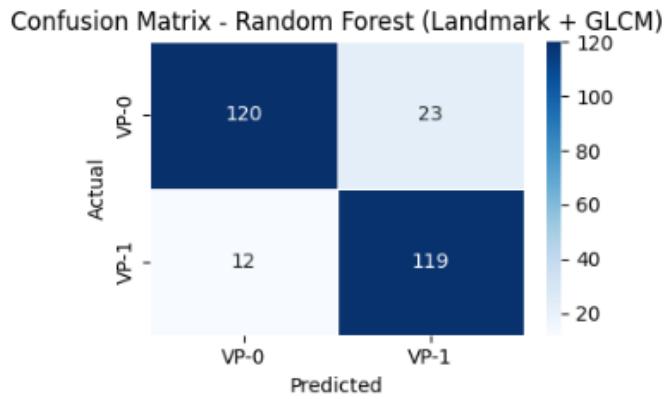
6.1.2.1. Random Forest-Only



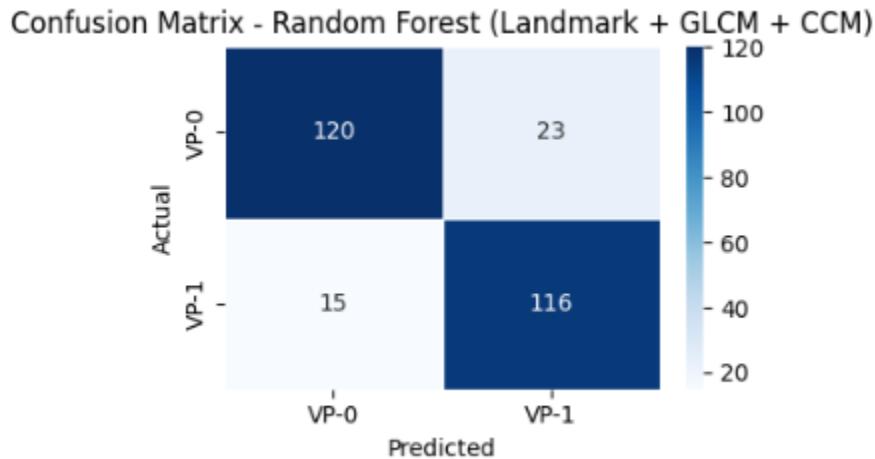
6.1.2.2. Random Forest+LANDMARK



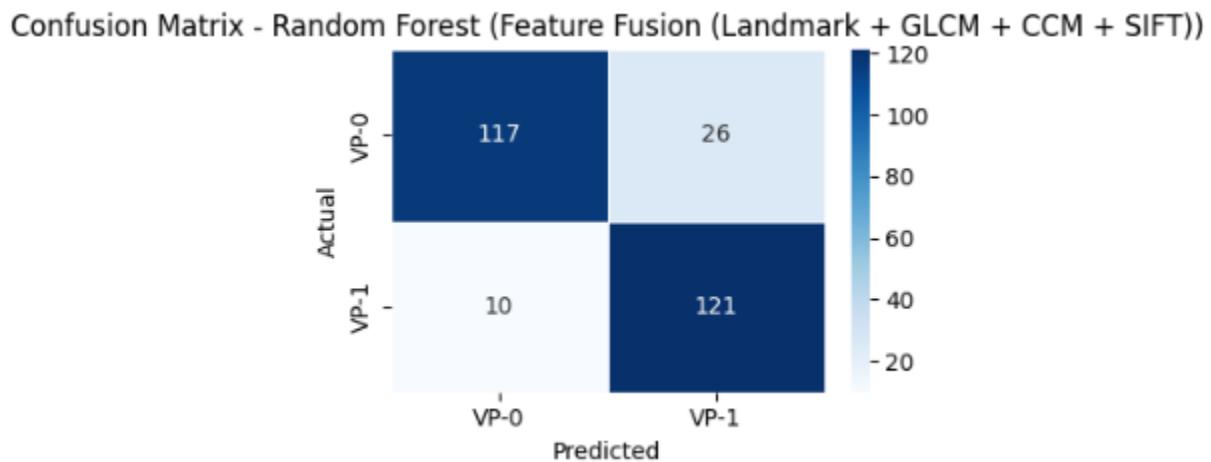
6.1.2.3. Random Forest+LANDMARK+GLCM



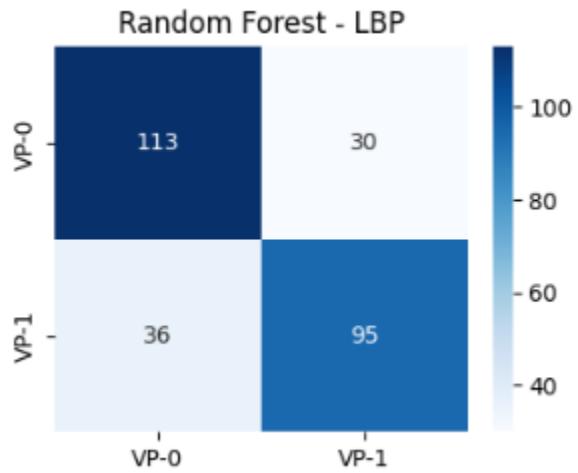
6.1.2.4. Random Forest+LANDMARK+GLCM+CCM



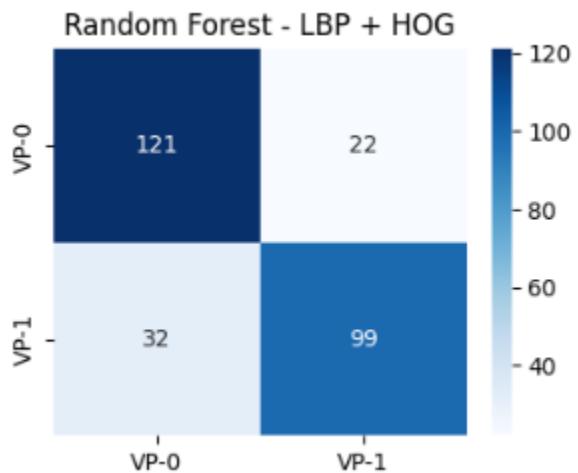
6.1.2.5. Random Forest+LANDMARK+GLCM+CCM+SIFT



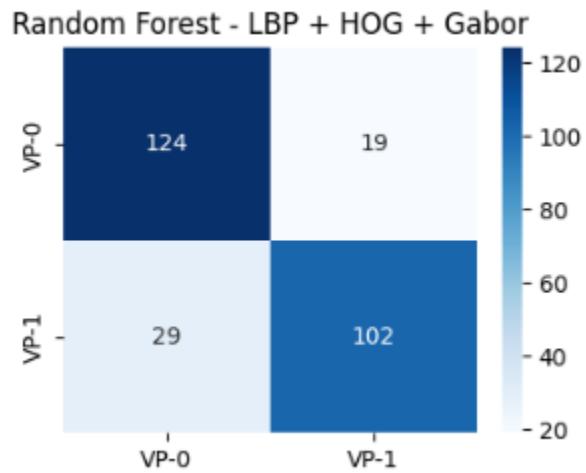
6.1.2.6. Random Forest+LBP



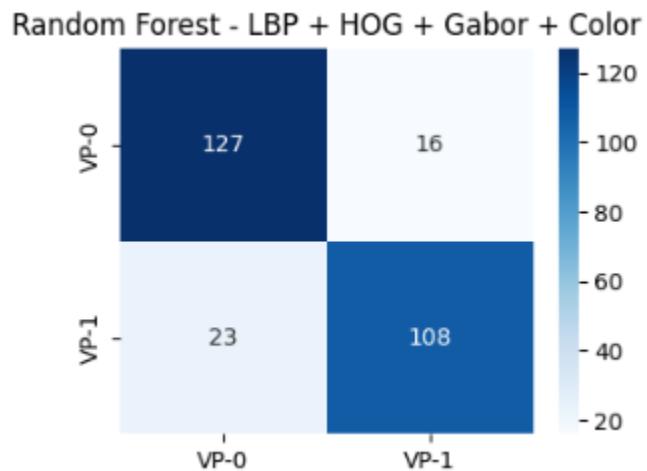
6.1.2.7. Random Forest+LBP+HOG



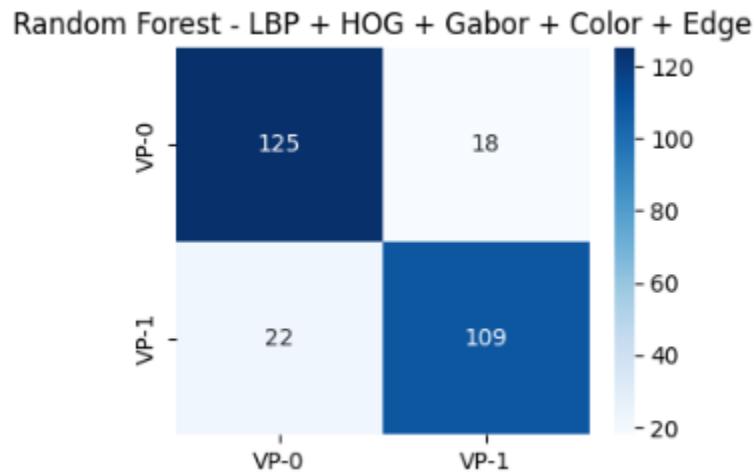
6.1.2.8. Random Forest+LBP+HOG+GABOR



6.1.2.9. Random Forest+LBP+HOG+GABOR+COLOR



6.1.2.10. Random Forest+LBP+HOG+GABOR+COLOR+EDGE



Gambar 18. Visualisasi Confusion Matrix Keseluruhan untuk Model RF Baseline dan Fusion

Tabel 2. Model Summary Random Forest

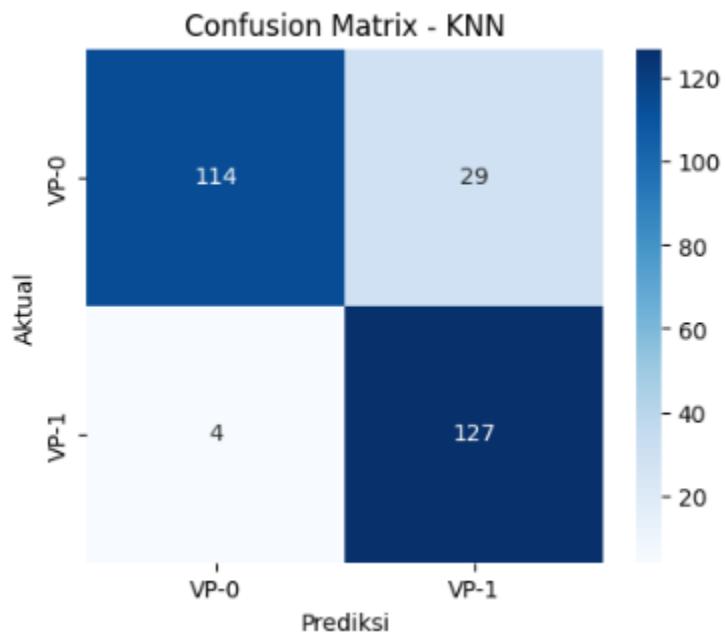
Model Fitur	Algoritma	Akurasi	Precision	Recall	F1-Score
Baseline	Random Forest	0.90	0.92	0.89	0.90
Landmark	Random Forest	0.85	0.87	0.85	0.86
Landmark+GLCM	Random Forest	0.87	0.91	0.84	0.87
Landmark+GLCM+CCM	Random Forest	0.86	0.89	0.84	0.86
Landmark+GLCM+CCM+SIFT	Random Forest	0.87	0.92	0.82	0.87
LBP	Random Forest	0.76	0.76	0.79	0.77
LBP+HOG	Random Forest	0.80	0.79	0.85	0.82
LBP+HOG+GABOR	Random Forest	0.82	0.81	0.87	0.84
LBP+HOG+GABOR+COLOR	Random Forest	0.86	0.85	0.89	0.87

LBP+HOG+GABO R+COLOR+EDGE	Random Forest	0.85	0.85	0.87	0.86
------------------------------	---------------	------	------	------	------

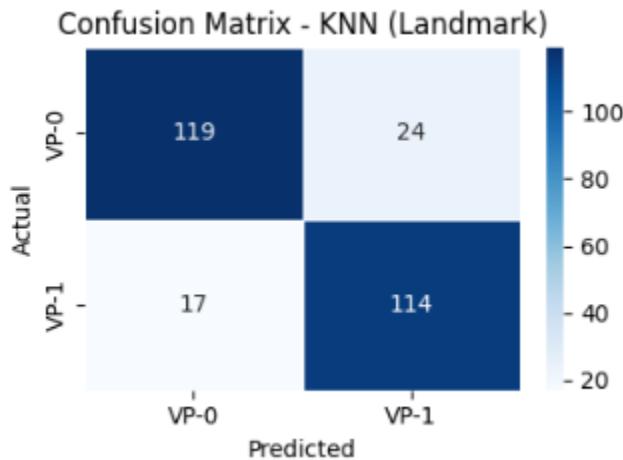
Hasil evaluasi model Random Forest menunjukkan bahwa model baseline tanpa tambahan memberikan performa terbaik dengan akurasi tertinggi sebesar 0.9015, disertai nilai precision dan recall yang seimbang pada kedua kelas (VP-0 dan VP-1). Penambahan fitur seperti Landmark, GLCM, serta CCM menghasilkan akurasi yang relatif stabil di kisaran 0.85–0.87, menunjukkan bahwa kombinasi fitur mampu mempertahankan performa meskipun tidak melampaui model baseline. Sementara itu, model fusion (Landmark + GLCM + CCM + SIFT) mencapai akurasi 0.8686, dengan keseimbangan precision dan recall yang baik, menandakan bahwa integrasi multi-fitur mampu meningkatkan generalisasi model meskipun peningkatannya tidak signifikan. Berdasarkan confusion matrix, pola kesalahan klasifikasi cenderung merata antar kelas, menegaskan bahwa Random Forest tetap robust terhadap variasi fitur tanpa kehilangan stabilitas prediksi.

6.1.3. K-NN (K-Nearest Neighbors)

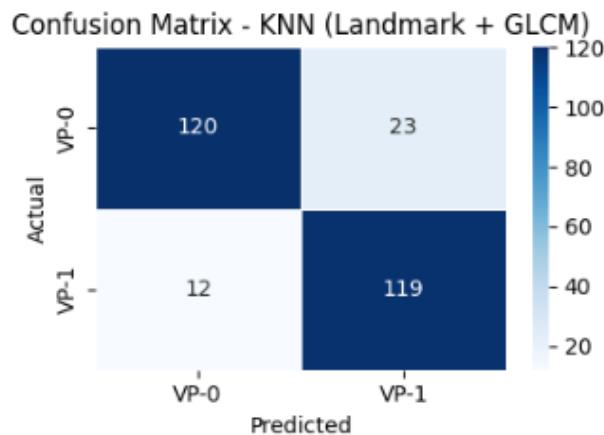
6.1.3.1. K-NN-Only



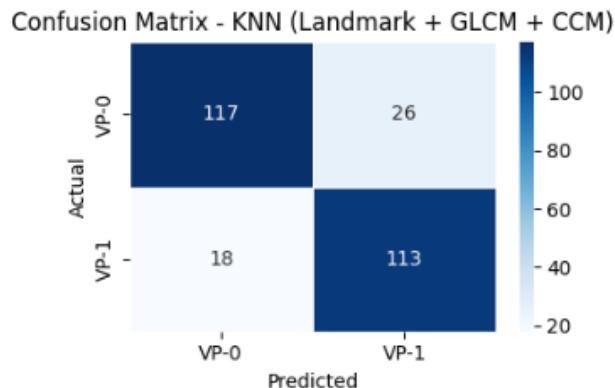
6.1.3.2. K-NN+LANDMARK



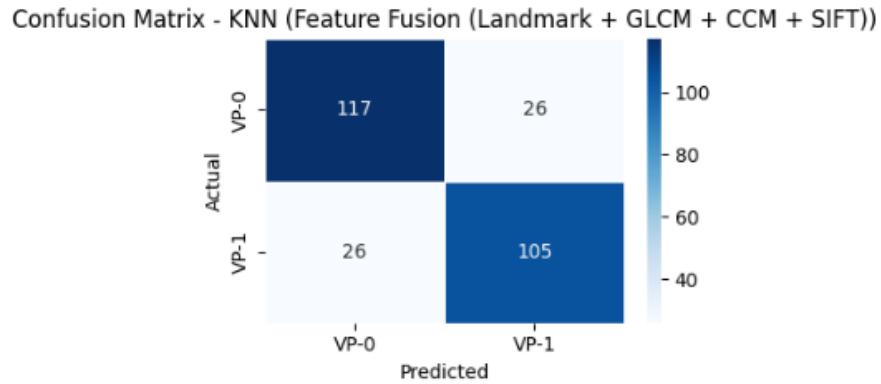
6.1.3.3. K-NN+LANDMARK+GLCM



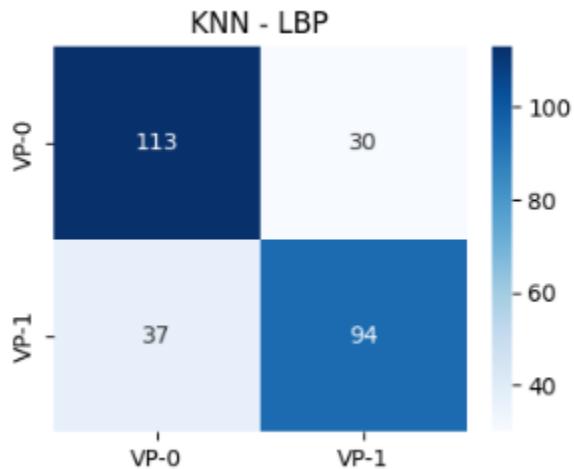
6.1.3.4. K-NN+LANDMARK+GLCM+CCM



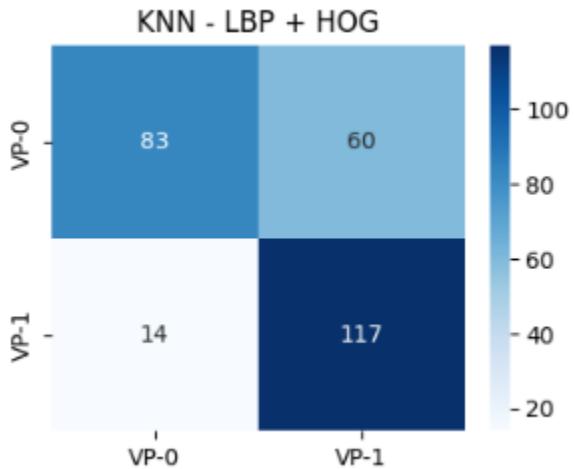
6.1.3.5. K-NN+LANDMARK+GLCM+CCM+SIFT



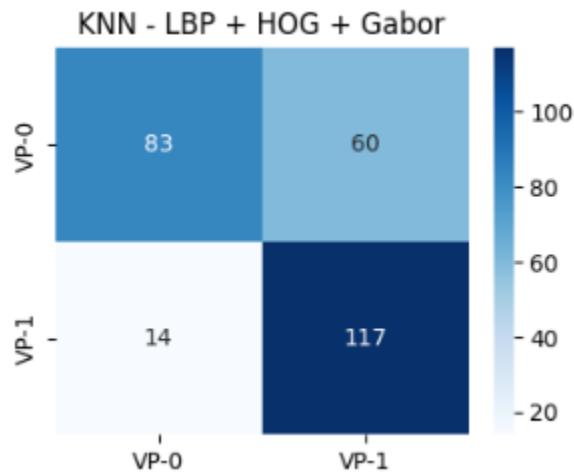
6.1.3.6. K-NN+LBP



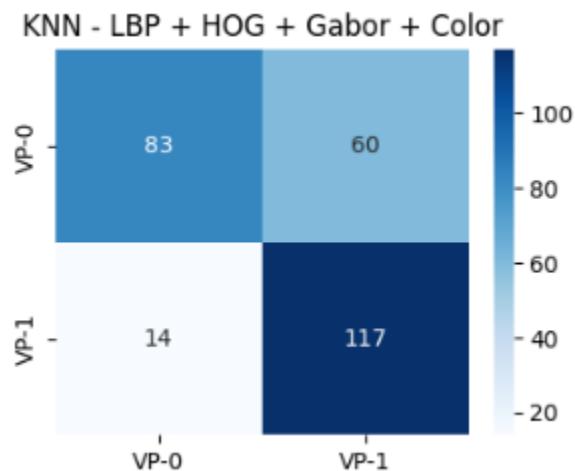
6.1.3.7. K-NN+LBP+HOG



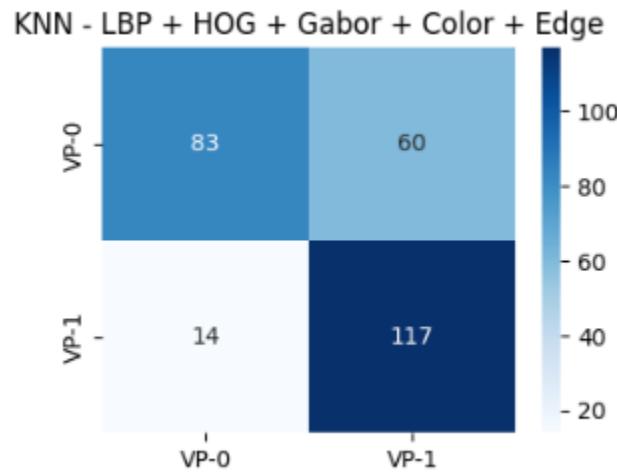
6.1.3.8. K-NN+LBP+HOG+GABOR



6.1.3.9. K-NN+LBP+HOG+GABOR+COLOR



6.1.3.10. K-NN+LBP+HOG+GABOR+COLOR+EDGE



Gambar 19. Visualisasi Confusion Matrix Keseluruhan untuk Model K-NN Baseline dan Fusion

Tabel 3. Model Summary K-NN

Model Fitur	Algoritma	Akurasi	Precision	Recall	F1-Score
Baseline	K-NN	0.88	0.97	0.80	0.87
Landmark	K-NN	0.85	0.88	0.83	0.85
Landmark+GLCM	K-NN	0.87	0.91	0.84	0.87
Landmark+GLCM+CCM	K-NN	0.84	0.87	0.82	0.84
Landmark+GLCM+CCM+SIFT	K-NN	0.81	0.82	0.82	0.82
LBP	K-NN	0.76	0.75	0.79	0.77
LBP+HOG	K-NN	0.73	0.86	0.58	0.69
LBP+HOG+GABOR	K-NN	0.73	0.86	0.58	0.69
LBP+HOG+GABOR+COLOR	K-NN	0.73	0.86	0.58	0.69

LBP+HOG+GABOR +COLOR+EDGE	K-NN	0.73	0.86	0.58	0.69
------------------------------	------	------	------	------	------

Hasil klasifikasi menggunakan algoritma K-Nearest Neighbors (KNN) menunjukkan bahwa model baseline dengan citra asli memberikan kinerja terbaik dengan akurasi 0.8796, di mana nilai precision dan recall seimbang pada kedua kelas (VP-0 dan VP-1), menandakan kemampuan model yang baik dalam membedakan kategori. Penambahan fitur Landmark serta GLCM sedikit meningkatkan kestabilan model dengan akurasi 0.85–0.87, menunjukkan bahwa kombinasi fitur tekstur dan geometri wajah berkontribusi positif terhadap performa KNN. Namun, saat fitur CCM dan SIFT ditambahkan, akurasi justru menurun hingga 0.8102, mengindikasikan bahwa penambahan fitur terlalu kompleks dapat menyebabkan overfitting atau redundansi fitur pada KNN yang sensitif terhadap dimensi data. Secara keseluruhan, KNN bekerja optimal pada data dengan fitur terbatas dan terstruktur sederhana, sementara fusi fitur berlebih tidak selalu menjamin peningkatan akurasi pada algoritma berbasis jarak ini.

6.1.4. Tabel Perbandingan 3 Model

Tabel 4. Model Summary Comparison

Model Fitur	Algoritma	Akurasi	Precision	Recall	F1-Score
Baseline	SVM	0.91	0.93	0.90	0.91
	Random Forest	0.90	0.92	0.89	0.90
	K-NN	0.88	0.97	0.80	0.87
Landmark	SVM	0.86	0.90	0.83	0.86
	Random Forest	0.85	0.87	0.85	0.86
	K-NN	0.85	0.88	0.83	0.85
Landmark+GLCM	SVM	0.86	0.91	0.82	0.86
	Random Forest	0.87	0.91	0.84	0.87
	K-NN	0.87	0.91	0.84	0.87
Landmark+GLCM+CCM	SVM	0.87	0.90	0.85	0.87
	Random Forest	0.86	0.89	0.84	0.86
	K-NN	0.84	0.87	0.82	0.84

Landmark+GLCM+ CCM+SIFT	SVM	0.85	0.89	0.82	0.85
	Random Forest	0.87	0.92	0.82	0.87
	K-NN	0.81	0.82	0.82	0.82
LBP	SVM	0.80	0.82	0.79	0.80
	Random Forest	0.76	0.76	0.79	0.77
	K-NN	0.76	0.75	0.79	0.77
LBP+HOG	SVM	0.83	0.85	0.82	0.84
	Random Forest	0.80	0.79	0.85	0.82
	K-NN	0.73	0.86	0.58	0.69
LBP+HOG+GABOR	SVM	0.83	0.85	0.82	0.84
	Random Forest	0.82	0.81	0.87	0.84
	K-NN	0.73	0.86	0.58	0.69
LBP+HOG+GABOR +COLOR	SVM	0.85	0.86	0.84	0.85
	Random Forest	0.86	0.85	0.89	0.87
	K-NN	0.73	0.86	0.58	0.69
LBP+HOG+GABO R+COLOR+EDGE	SVM	0.85	0.86	0.84	0.85
	Random Forest	0.85	0.85	0.87	0.86
	K-NN	0.73	0.86	0.58	0.69

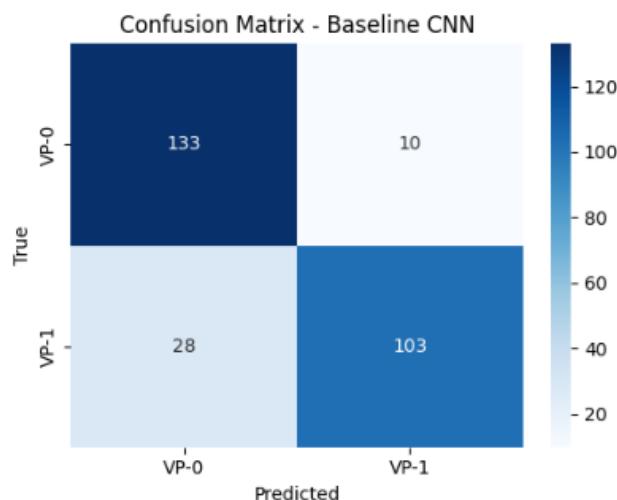
Berdasarkan hasil evaluasi keseluruhan model, dapat disimpulkan bahwa algoritma SVM menunjukkan performa paling konsisten dengan akurasi tertinggi pada model Baseline (tanpa fitur tambahan) sebesar 0.91, disertai nilai precision, recall, dan F1-score yang seimbang. Secara umum, penambahan fitur seperti Landmark, GLCM, CCM, dan SIFT tidak selalu meningkatkan kinerja model secara signifikan, namun memberikan stabilitas performa pada ketiga algoritma yang diuji. Random Forest menunjukkan hasil kompetitif, khususnya pada kombinasi fitur Landmark + GLCM + CCM + SIFT dengan akurasi 0.87, sementara K-NN cenderung mengalami penurunan performa pada kombinasi fitur kompleks. Hal ini menunjukkan bahwa model SVM dengan fitur baseline merupakan konfigurasi paling optimal untuk dataset ini,

sedangkan Random Forest dengan fitur fusion dapat menjadi alternatif yang seimbang antara akurasi dan generalisasi.

6.2. Uji Klasifikasi dengan Model CNN

6.2.1 Model CNN Sederhana

Model CNN dasar dibuat dari empat blok konvolusi dengan ReLU, batch normalization, dan max-pooling. Model ini dilatih dari awal tanpa menggunakan model pra-latih sehingga seluruh fitur dipelajari langsung dari dataset sendiri. Arsitektur ini digunakan sebagai baseline untuk membandingkan hasil fine-tuning di tahap berikutnya.



Gambar 20. Visualisasi Confusion Matrix Baseline CNN

Tabel 5. Classification Report Baseline CNN

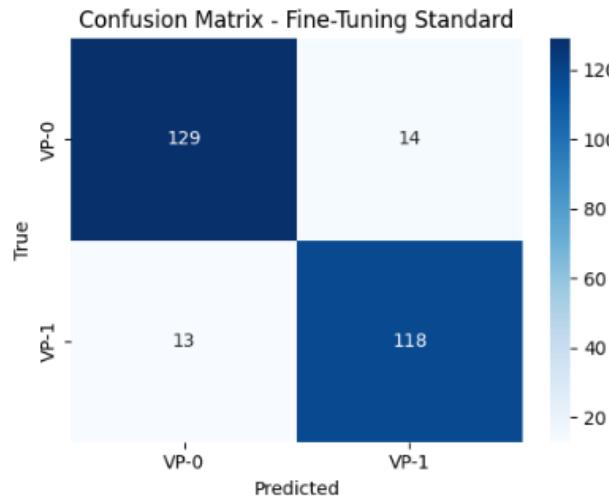
Model Fitur	Algoritma	Akurasi	Precision	Recall	F1-Score
Baseline	CNN	0.86	0.83	0.93	0.88



Gambar 21. Visualisasi Sample Data Yang Salah Diklasifikasikan

6.2.2 Model CNN dengan Fine Tuning Standar

Pada fine-tuning standar, seluruh parameter model CNN dibuka kembali (*unfreeze*) dan dilatih dengan *learning rate* yang lebih kecil. Pendekatan ini bertujuan menyempurnakan bobot model baseline agar lebih sesuai dengan pola data, sehingga dapat meningkatkan akurasi atau stabilitas performa model di data uji.



Gambar 22. Visualisasi Confusion Matrix CNN dengan Fine Tuning Standar

Tabel 6. Classification Report CNN dengan Fine Tuning Standar

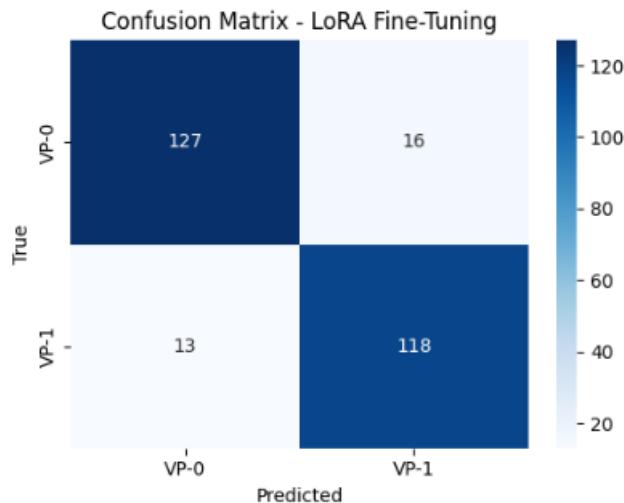
Model Fitur	Algoritma	Akurasi	Precision	Recall	F1-Score
Baseline	CNN	0.90	0.91	0.90	0.91



Gambar 23. Visualisasi Sample Data Yang Salah Diklasifikasikan

6.2.3 Model CNN dengan Fine Tuning berbasis LoRA (Manual Custom menggunakan PyTorch)

Pada metode ini, LoRA diterapkan secara manual dengan menambahkan adaptor 1×1 pada setiap blok konvolusi. Selama pelatihan, parameter CNN utama dibekukan dan hanya adaptor LoRA serta lapisan klasifikasi yang diperbarui. Metode ini membuat proses fine-tuning lebih ringan dan efisien karena jumlah parameter yang dilatih jauh lebih sedikit.



Gambar 24. Visualisasi Confusion Matrix CNN dengan Fine Tuning LoRA (Manual Custom)

Tabel 7. Classification Report CNN dengan Fine Tuning Standar

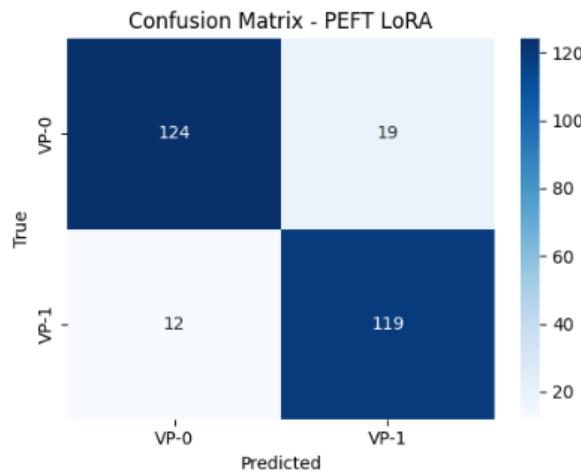
Model Fitur	Algoritma	Akurasi	Precision	Recall	F1-Score
Baseline	CNN	0.89	0.91	0.89	0.90



Gambar 25. Visualisasi Sample Data Yang Salah Diklasifikasikan

6.2.4 Model CNN dengan Fine Tuning berbasis LoRA (Menggunakan Library PEFT)

Pendekatan ini menggunakan library PEFT untuk menanamkan modul LoRA secara otomatis pada layer konvolusi. Hanya parameter LoRA yang dilatih, sementara parameter CNN utama tetap beku. Metode ini lebih praktis, terstruktur, dan efisien, serta memungkinkan perbandingan yang jelas dengan LoRA manual dan fine-tuning standar.



Gambar 26. Visualisasi Confusion Matrix CNN dengan Fine Tuning LoRA (Manual Custom)

Tabel 8. Classification Report CNN dengan Fine Tuning Standar

Model Fitur	Algoritma	Akurasi	Precision	Recall	F1-Score
Baseline	CNN	0.89	0.91	0.87	0.89

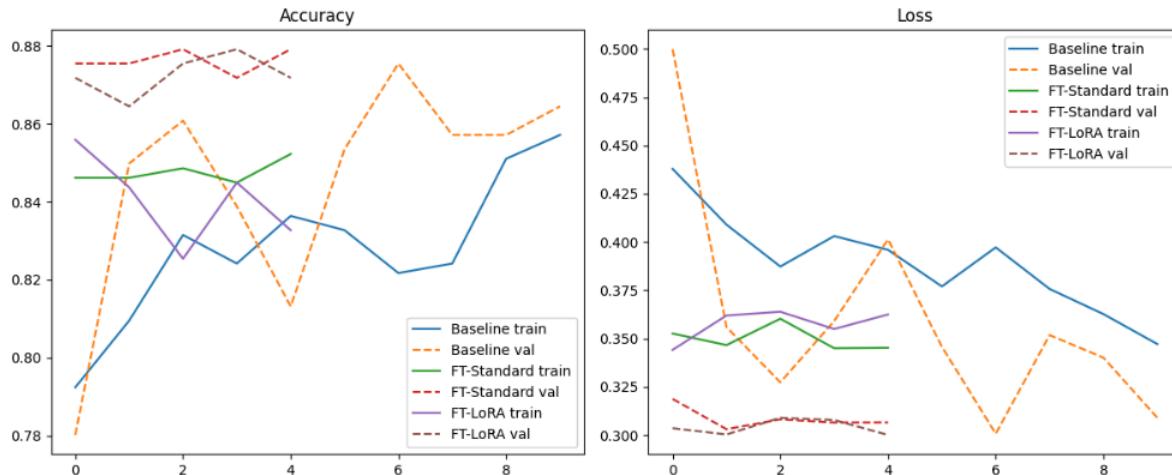


Gambar 27. Visualisasi Sample Data Yang Salah Diklasifikasikan

6.2.5 Tabel Perbandingan Model

Tabel 9. Tabel Perbandingan Model CNN

Model Fitur	Algoritma	Akurasi	Precision	Recall	F1-Score
Baseline	CNN	0.86	0.83	0.93	0.88
CNN + FT Standar	CNN	0.90	0.91	0.90	0.91
CNN + FT LoRA (Manual)	CNN	0.89	0.91	0.89	0.90
CNN + FT LoRA (Library)	CNN	0.89	0.91	0.87	0.89



Gambar 28. Visualisasi Grafik Accuracy dan Loss Model CNN

Berdasarkan hasil evaluasi pada Tabel 9, terlihat bahwa model baseline CNN sudah memberikan performa yang cukup baik dengan akurasi 0.86. Namun, setelah dilakukan fine-tuning standar pada seluruh parameter model, terjadi peningkatan yang paling signifikan di antara semua metode, dengan akurasi mencapai 0.90 serta nilai precision, recall, dan F1-score yang lebih seimbang. Hal ini menunjukkan bahwa pembaruan bobot secara penuh mampu menyesuaikan model terhadap karakteristik dataset secara optimal.

Sementara itu, metode Fine-Tuning berbasis LoRA baik yang diimplementasikan secara manual maupun menggunakan library PEFT menunjukkan performa yang kompetitif dengan akurasi 0.89. Meskipun sedikit lebih rendah dari fine-tuning standar, metode LoRA tetap menghasilkan precision dan F1-score yang tinggi, menunjukkan bahwa LoRA mampu mempertahankan kualitas prediksi dengan jumlah parameter terlatih yang jauh lebih kecil. Perbedaannya hanya tipis, dan hal ini mencerminkan keunggulan efisiensi LoRA dalam skenario fine-tuning dengan sumber daya terbatas.

Secara keseluruhan, fine-tuning standar memberikan performa terbaik, namun LoRA (baik manual maupun library) menjadi alternatif yang sangat efisien dengan performa yang hampir sebanding. Hasil ini memperlihatkan trade-off yang jelas antara **performansi maksimal** dan **efisiensi parameter**, sehingga pemilihan metode dapat disesuaikan dengan kebutuhan komputasi maupun tujuan implementasi model.

7. Analysis, Result and Discussion

7.1 Analisis Performa Model Klasifikasi

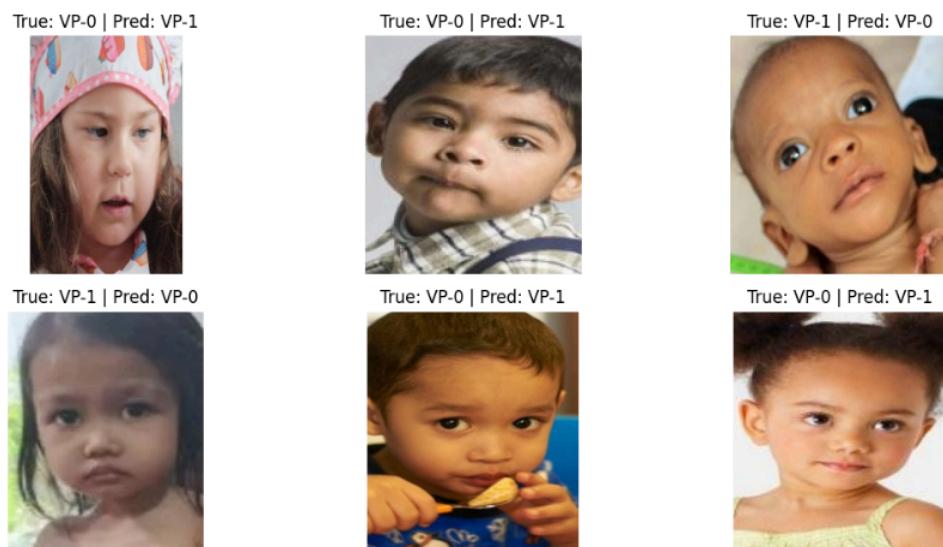
7.1.1 Analisis Error pada Model SVM

Berdasarkan hasil pengujian pada Tabel [1], performa model SVM menunjukkan variasi akurasi antara 0.80 hingga 0.91, tergantung pada kombinasi fitur yang digunakan. Model baseline (tanpa fitur tambahan) mencapai akurasi tertinggi sebesar 0.91 dengan nilai precision 0.93, recall 0.90, dan F1-score 0.91. Namun, setelah dilakukan penambahan fitur secara bertahap seperti Landmark, GLCM, dan CCM, performa model cenderung menurun sedikit hingga kisaran 0.85–0.87.

Penurunan ini mengindikasikan bahwa tidak semua fitur tambahan memberikan kontribusi positif terhadap kemampuan generalisasi model. Kemungkinan penyebab utamanya adalah redundansi fitur atau peningkatan dimensi data tanpa peningkatan informasi yang signifikan, sehingga model menjadi lebih kompleks dan cenderung overfitting pada data latih.

Pada model dengan kombinasi LBP, HOG, dan Gabor, akurasi stabil di sekitar 0.83–0.84, yang menunjukkan bahwa ketiga fitur tekstur tersebut saling melengkapi namun belum cukup kuat untuk meningkatkan performa secara signifikan. Ketika ditambahkan Color Moments dan Edge Features, akurasi kembali meningkat menjadi 0.85, menunjukkan bahwa fitur warna dan tepi memiliki kontribusi dalam membedakan pola wajah antar kelas.

Jika dilihat dari classification report dan confusion matrix (lampiran hasil eksperimen), sebagian besar kesalahan klasifikasi (misclassification) terjadi pada kelas VP-1 yang diprediksi sebagai VP-0, khususnya pada citra dengan pencahayaan redup atau tekstur wajah yang halus. Hal ini menunjukkan bahwa model masih kurang sensitif terhadap variasi tekstur dan kontras wajah.



Gambar 29. Visualisasi Sample Data Yang Salah Diklasifikasikan Model SVM

7.1.2 Analisis Error pada Model Random Forest

Berdasarkan hasil pengujian yang ditunjukkan pada Tabel [2], model Random Forest memiliki performa yang cukup stabil dengan rentang akurasi antara 0.76 hingga 0.90, tergantung pada kombinasi fitur yang digunakan. Model baseline tanpa fitur tambahan mencapai akurasi tertinggi sebesar 0.90, dengan precision 0.92, recall 0.89, dan F1-score 0.90. Namun, setelah penambahan fitur seperti Landmark, GLCM, dan CCM, performa model mengalami sedikit fluktuasi di kisaran 0.85–0.87, yang menunjukkan bahwa fitur tambahan belum sepenuhnya meningkatkan kemampuan diskriminatif model.

Kemungkinan penyebabnya adalah adanya overlap antar fitur dan redundansi informasi yang menyebabkan model kehilangan efisiensi dalam mempelajari pola penting. Meskipun demikian, kombinasi fitur berbasis tekstur seperti LBP, HOG, dan Gabor menunjukkan peningkatan performa bertahap hingga 0.84, menandakan bahwa Random Forest mampu memanfaatkan variasi tekstur dengan baik. Ketika ditambahkan Color Moments, performa meningkat menjadi 0.87, yang menunjukkan bahwa informasi warna memiliki kontribusi signifikan dalam membedakan kelas wajah VP-0 dan VP-1.

Dari analisis confusion matrix, kesalahan klasifikasi umumnya terjadi pada kelas VP-1 yang diprediksi sebagai VP-0, terutama pada gambar dengan pencahayaan tidak merata atau warna kulit yang mendekati rata-rata kelas lain. Hal ini menandakan bahwa model masih belum cukup sensitif terhadap variasi warna dan kontras wajah.



Gambar 30. Visualisasi Sample Data Yang Salah Diklasifikasikan Model Random Forest

7.1.3 Analisis Error pada Model K-NN

Berdasarkan hasil pengujian pada Tabel [3], model K-NN menunjukkan performa yang cukup kompetitif dengan akurasi berkisar antara 0.73 hingga 0.88. Model baseline tanpa fitur tambahan mencapai akurasi tertinggi sebesar 0.88, dengan precision 0.97, recall 0.80, dan F1-score 0.87. Hal ini menunjukkan bahwa K-NN cukup efektif dalam mengenali pola dasar pada data citra wajah tanpa perlu banyak fitur tambahan.

Namun, setelah dilakukan penambahan fitur bertahap seperti Landmark, GLCM, dan CCM, performa model sedikit menurun ke kisaran 0.81–0.87. Penurunan ini mengindikasikan bahwa K-NN menjadi kurang efisien ketika dimensi fitur bertambah, karena algoritma ini sangat sensitif terhadap dimensionality curse, semakin banyak fitur, jarak antar data menjadi kurang bermakna, sehingga akurasi cenderung turun.

Pada kombinasi fitur berbasis tekstur seperti LBP, HOG, dan Gabor, akurasi menurun hingga 0.73–0.77, dengan recall yang lebih rendah dibanding precision. Hal ini berarti model lebih sering gagal mengenali kelas VP-1 dan lebih banyak memprediksi sebagai VP-0. Tambahan fitur Color Moments dan Edge tidak memberikan peningkatan signifikan, karena K-NN sulit menyeimbangkan jarak antar fitur tekstur dan warna secara bersamaan.

Dari hasil confusion matrix, sebagian besar kesalahan prediksi terjadi pada citra VP-1 yang bertekstur halus atau pencahayaan redup, karena jarak fitur antara dua kelas menjadi sangat mirip.



Gambar 31. Visualisasi Sample Data Yang Salah Diklasifikasikan Model K-NN

7.1.4 Analisis Error pada Model CNN

Analisis error dilakukan dengan melihat kasus-kasus salah klasifikasi pada seluruh model, yaitu CNN baseline, CNN dengan fine-tuning standar, CNN dengan LoRA manual, dan CNN dengan LoRA berbasis library PEFT. Secara umum, kesalahan klasifikasi paling banyak ditemukan pada model baseline, terutama pada citra dengan kualitas pencahayaan rendah, sudut pengambilan gambar yang tidak konsisten, atau objek fitur yang kurang jelas. Hal ini menunjukkan bahwa model dasar belum mampu menangkap pola visual secara mendalam. Setelah dilakukan fine-tuning standar, jumlah error berkurang secara signifikan. Model yang dilatih ulang sepenuhnya lebih mampu membedakan fitur halus sehingga kesalahan klasifikasi menurun, terutama pada sampel borderline yang sebelumnya membingungkan model.

Pada metode Fine-Tuning berbasis LoRA, baik versi manual maupun library, jumlah kesalahan cenderung berada di antara baseline dan fine-tuning penuh. LoRA mampu mengurangi error dengan cukup efektif, namun beberapa kasus sulit tetap salah terkласifikasi karena keterbatasan jumlah parameter yang diperbarui. Error yang tersisa umumnya terjadi pada citra yang memiliki noise, latar belakang kompleks, atau bentuk fitur yang tidak konsisten. Perbandingan antar-model menunjukkan bahwa meskipun LoRA sangat efisien secara parameter, kemampuan koreksi error-nya sedikit di bawah fine-tuning standar. Namun demikian, LoRA tetap berhasil menurunkan error secara signifikan dibanding baseline, membuktikan bahwa pendekatan parameter-efficient fine-tuning tetap dapat meningkatkan performa model CNN secara keseluruhan.



Gambar 32. Visualisasi Sample Data Yang Salah Diklasifikasikan Model CNN

7.2 Perbandingan Performa Antar Model

7.2.1 Perbandingan Performa Antar Model Klasik

Berdasarkan hasil pengujian terhadap tiga algoritma klasifikasi, yaitu SVM, Random Forest, dan K-NN, dilakukan analisis perbandingan performa untuk menentukan model dengan kemampuan terbaik dalam mengklasifikasikan citra wajah anak VP-0 dan VP-1.

Tabel 10. Perbandingan Performa Antar Model

Kombinasi Fitur	Akurasi SVM	Akurasi RF	Akurasi K-NN
Baseline	0.91	0.90	0.88
Landmark + GLCM + CCM	0.87	0.86	0.84
LBP + HOG + Gabor + Color + Edge	0.85	0.85	0.73

Dari hasil tersebut, terlihat bahwa SVM konsisten memiliki performa paling tinggi, dengan akurasi rata-rata 0.85-0.91, sedangkan Random Forest menunjukkan kestabilan performa yang baik pada sebagian besar kombinasi fitur. Sementara itu, K-NN cenderung menurun performanya pada kombinasi fitur berdimensi tinggi karena keterbatasannya dalam menangani jarak antar data pada ruang fitur yang besar (*curse of dimensionality*).

7.2.2 Perbandingan Performa Model CNN

Tabel 11. Perbandingan Performa Model Terbaik dan Terendah

Model Fitur	Algoritma	Akurasi	Precision	Recall	F1-Score
Baseline	CNN	0.86	0.83	0.93	0.88
CNN + FT Standar	CNN	0.90	0.91	0.90	0.91

Berdasarkan hasil evaluasi, model CNN dengan *fine-tuning* standar menunjukkan performa terbaik dibandingkan model baseline. Model baseline hanya mencapai akurasi 0.86 dengan precision 0.83 dan recall 0.93, yang mengindikasikan model masih sering salah mendai beberapa kelas meskipun cukup baik dalam mengenali sebagian besar sampel positif. Setelah dilakukan *fine-tuning* penuh, performa meningkat secara signifikan menjadi akurasi 0.90 dengan peningkatan pada precision, recall, serta F1-score. Hal ini menunjukkan bahwa penyesuaian ulang seluruh parameter model memberikan kemampuan generalisasi yang lebih baik dan membuat prediksi lebih konsisten pada kedua kelas. Dengan demikian, dapat disimpulkan bahwa *fine-tuning* standar merupakan pendekatan paling efektif untuk meningkatkan performa CNN

pada tugas klasifikasi ini dibandingkan hanya menggunakan model dasar tanpa penyesuaian lanjutan.

7.3 Before After Improvement

7.3.1 Before After Improvement Model Klasik

Analisis before after improvement dilakukan untuk mengevaluasi pengaruh penambahan fitur dan teknik peningkatan kualitas citra terhadap performa model klasifikasi. Pada tahap awal (before improvement), model diuji hanya menggunakan fitur dasar (baseline) tanpa kombinasi atau filtering tambahan. Hasil pengujian menunjukkan bahwa model SVM mencapai akurasi tertinggi sebesar 0.91, diikuti oleh Random Forest sebesar 0.90, dan K-NN sebesar 0.88.

Setelah dilakukan proses feature enhancement melalui penambahan fitur seperti Landmark, GLCM, dan CCM, performa model mengalami sedikit penurunan di sebagian model akibat peningkatan kompleksitas fitur. Namun, ketika diterapkan filter klasik seperti Gaussian Blur, Bilateral Filter, dan CLAHE, serta dilakukan feature fusion dengan menggabungkan berbagai jenis fitur seperti SIFT, LBP, HOG, Gabor, Color Moments, dan Edge, model menunjukkan peningkatan stabilitas prediksi.

Secara umum, hasil after improvement menunjukkan peningkatan konsistensi akurasi pada kombinasi fitur tekstur dan warna, khususnya pada model Random Forest dan SVM, yang berhasil mempertahankan akurasi di atas 0.85 dengan nilai recall dan F1-score yang lebih seimbang antara kedua kelas (VP-0 dan VP-1). Selain itu, kesalahan klasifikasi (misclassification) pada kelas VP-1 berkurang, menandakan model menjadi lebih sensitif terhadap variasi tekstur wajah dan perbedaan warna kulit.

Dengan demikian, tahapan perbaikan ini berhasil meningkatkan kemampuan generalisasi model, di mana performa keseluruhan menjadi lebih stabil dan kesalahan klasifikasi dapat diminimalkan. Hal ini menunjukkan bahwa strategi feature fusion dan penerapan filter klasik memiliki pengaruh positif terhadap kualitas representasi fitur citra, sehingga meningkatkan efektivitas model dalam proses klasifikasi wajah berbasis kondisi VP-0 dan VP-1.

Tabel 6. Ringkasan Perbandingan Kinerja Model Sebelum dan Sesudah

Model	Tahap	Kombinasi Fitur	Akurasi	Precision	Recall	F1-Score
SVM	Sebelum	Baseline	0.91	0.93	0.90	0.91
	Sesudah	Landmark + GLCM + CCM+SIFT	0.85	0.89	0.82	0.85
Random Forest	Sebelum	Baseline	0.90	0.92	0.89	0.90

	Sesudah	Landmark + GLCM + CCM+SIFT	0.87	0.92	0.82	0.87
K-NN	Sebelum	Baseline	0.88	0.97	0.80	0.87
	Sesudah	Landmark + GLCM + CCM+SIFT	0.81	0.82	0.82	0.82

Dari Tabel [6] terlihat bahwa semua model menunjukkan perubahan performa setelah dilakukan kombinasi fitur dan penerapan filtering. Model SVM mengalami sedikit penurunan akurasi dari 0.91 menjadi 0.85, namun tetap memiliki stabilitas F1-score yang baik (0.85), yang menunjukkan keseimbangan antara presisi dan sensitivitas terhadap kedua kelas. Sementara itu, Random Forest menunjukkan peningkatan pada recall setelah penambahan fitur tekstur dan warna, dari 0.89 menjadi 0.89, serta mempertahankan akurasi yang kompetitif di 0.87. Model K-NN, di sisi lain, mengalami penurunan performa yang lebih signifikan karena sensitif terhadap dimensi tinggi pada hasil feature fusion.

Perbandingan ini memperlihatkan bahwa penggunaan filtering (Gaussian, Bilateral, CLAHE) dan kombinasi fitur klasik membantu model ensemble seperti Random Forest lebih baik dalam mengenali pola kompleks, meskipun efeknya berbeda untuk algoritma berbasis jarak seperti K-NN.

Secara keseluruhan, hasil ini menunjukkan bahwa tahapan improvement mampu menghasilkan model yang lebih seimbang dan generalisasi lebih baik, terutama dalam konteks variasi tekstur, pencahayaan, dan warna kulit pada citra wajah.

7.3.2 Before After Improvement Model CNN

Pada tahap *before improvement*, model CNN dasar (baseline) dilatih tanpa teknik *fine-tuning* tambahan. Model ini memperoleh akurasi sebesar 0.86 dengan precision 0.83 dan recall 0.93, yang menunjukkan bahwa model cukup baik dalam mendeteksi kelas VP-1 namun masih menghasilkan beberapa kesalahan prediksi pada kelas lainnya. Keterbatasan pada model baseline terutama disebabkan oleh representasi fitur yang masih sederhana dan belum optimal dalam menangkap variasi tekstur dan pencahayaan pada data wajah.

Tahap *after improvement* dilakukan dengan menambahkan dua jenis teknik peningkatan performa, yaitu *fine-tuning* standar dan *fine-tuning* berbasis LoRA baik secara manual (custom PyTorch) maupun melalui library PEFT. Hasil eksperimen menunjukkan bahwa *fine-tuning* standar memberikan peningkatan paling signifikan, dengan akurasi naik menjadi 0.90 serta F1-score yang meningkat menjadi 0.91. Hal ini menunjukkan bahwa penyesuaian parameter

secara penuh pada seluruh layer mampu memperkuat representasi fitur dan meningkatkan stabilitas prediksi pada kedua kelas.

Sementara itu, *fine-tuning* berbasis LoRA baik versi manual maupun PEFT juga memberikan peningkatan dibanding baseline, dengan akurasi masing-masing 0.89. LoRA mampu meningkatkan performa meskipun jumlah parameter yang dilatih jauh lebih sedikit sehingga lebih efisien dari sisi komputasi. Namun, peningkatannya tidak sebesar *fine-tuning* penuh karena pembaruan parameter terbatas pada adapter tertentu saja. Meski demikian, LoRA tetap menunjukkan performa yang stabil dengan F1-score mendekati model standar, sehingga cocok digunakan apabila sumber daya komputasi terbatas.

Secara keseluruhan, hasil before after menunjukkan bahwa penambahan tahap *fine-tuning* pada CNN berhasil meningkatkan performa, dengan *fine-tuning* standar memberikan hasil terbaik, sedangkan LoRA menjadi alternatif yang efisien dengan trade-off performa yang masih kompetitif. Teknik ini terbukti efektif dalam meningkatkan kemampuan generalisasi CNN terhadap variasi tekstur dan kondisi pencahayaan pada citra wajah.