

Perbandingan Kinerja Multilayer Perceptron Berbasis Fitur Ekstraksi dan DenseNet-121 pada Klasifikasi Buah Kakao

Performance Comparison of Multilayer Perceptron Based on Feature Extraction and DenseNet-121 in Cocoa Fruit Classification

Aulia Zahra Ramadhani^{*1}, Nurhikma Arifin², Chairi Nur Insani³

Program Studi Informatika, Universitas Sulawesi Barat, Majene, Sulawesi Barat

E-mail : auliazhraramdhni@gmail.com^{*1}, nurhikma_arifin@unsulbar.ac.id²,

chairini@unsulbar.ac.id³

**Corresponding author*

Received 11 March 2026; Revised 2 April 2026; Accepted 3 April 2026

Abstrak - Tanaman kakao (*Theobroma cacao*) merupakan komoditas penting bagi perekonomian Indonesia, namun produktivitasnya menurun akibat serangan penyakit dan hama seperti *black pod rot*, *pod borer*, dan *helopeltis*. Penelitian ini bertujuan membandingkan kinerja *Multilayer Perceptron* (MLP) dengan kombinasi fitur warna (HSV, L*a*b) dan tekstur (GLCM, LBP) terhadap DenseNet-121 untuk klasifikasi empat kondisi buah kakao yaitu *healthy*, *black pod rot*, *pod borer* dan *helopeltis*. Dataset primer 404 citra dikumpulkan dari perkebunan kakao di Sulawesi Barat. Tahapan penelitian meliputi *preprocessing*, segmentasi, ekstraksi fitur, dan klasifikasi menggunakan MLP dengan 15 kombinasi fitur serta DenseNet-121 dengan *transfer learning*. Kedua model menerapkan *early stopping* dan dievaluasi menggunakan akurasi dengan tiga *random seed* berbeda untuk memastikan validitas hasil yang robust. Hasil menunjukkan DenseNet-121 mencapai akurasi tertinggi 92,19% pada *learning rate* 0,001 dengan *seed* 43, sedangkan MLP terbaik pada kombinasi HSV+GLCM+L*a*b hanya 79,69% pada *learning rate* 0,001 dengan *seed* 99. Kombinasi fitur warna dan tekstur pada MLP konsisten mengungguli fitur tunggal. DenseNet-121 terbukti lebih unggul karena kemampuannya mengekstraksi fitur hierarkis otomatis melalui bobot *pre-trained* ImageNet, menjadikannya metode lebih efektif untuk klasifikasi penyakit kakao yang akurat.

Kata Kunci – DenseNet-121, Ekstraksi Fitur, Kakao, Klasifikasi Citra, *Multilayer Perceptron*

Abstract - Cocoa plant (*Theobroma cacao*) is an important commodity for the Indonesian economy, but its productivity has declined due to attacks by diseases and pests such as *black pod rot*, *pod borers*, and *Helopeltis*. This study aims to compare the performance of a *Multilayer Perceptron* (MLP) using a combination of color (HSV, L*a*b) and texture (GLCM, LBP) features against DenseNet-121 for classifying four cocoa fruit conditions: *healthy*, *black pod rot*, *pod borer*, and *helopeltis*. A primary dataset of 404 images was collected from cocoa plantations in West Sulawesi. The research stages included *preprocessing*, segmentation, feature extraction, and classification using an MLP with 15 feature combinations as well as DenseNet-121 with transfer learning. Both models employed *early stopping* and were evaluated using accuracy with three different random seeds to ensure robust validity of the results. The results show that DenseNet-121 achieved the highest accuracy of 92.19% at a learning rate of 0.001 with seed 43, while the best MLP performance with the HSV+GLCM+L*a*b feature combination was only 79.69% at a learning rate of 0.001 with seed 99. The combination of color and texture features in the MLP consistently outperformed single features. DenseNet-121 proved superior due to its ability to automatically extract hierarchical features through *pre-trained* ImageNet weights, making it a more effective method for accurate cocoa disease classification.

Keywords - Cacao, DenseNet-121, Feature Extraction, Image Processing, *Multilayer Perceptron*

1. PENDAHULUAN

Tanaman kakao (*Theobroma cacao*) memiliki peran penting dalam ekonomi Indonesia, yaitu sebagai penyumbang devisa non-migas serta sumber mata pencaharian bagi jutaan petani [1], [2], [3]. Namun, dalam beberapa tahun terakhir, produktivitas kakao mengalami penurunan signifikan akibat serangan penyakit [1], [4]. Serangan penyakit tidak hanya menurunkan kualitas biji kakao, tetapi juga meningkatkan biaya produksi [5]. Penurunan ini diperparah oleh mayoritas petani yang masih mengandalkan metode pengamatan konvensional yang memakan waktu dan rentan subjektivitas, sehingga petani khususnya di daerah terpencil yang minim pengetahuan sering kali mengalami kesulitan dalam membedakan gejala penyakit yang mirip secara akurat [3], [6].

Dengan berkembangnya teknologi kecerdasan buatan dan pengolahan citra digital, berbagai studi telah memanfaatkan ekstraksi fitur warna dan tekstur untuk meningkatkan akurasi klasifikasi. Misalnya, penelitian yang dilakukan oleh Basri dkk. dalam mengidentifikasi kekurangan kalium pada tanaman kakao dengan menggunakan MLP yang menggabungkan fitur LBP, GLCM, dan HSV, mencapai akurasi sebesar 93,33% [7]. Penelitian lain pada komoditas pertanian juga menunjukkan efektivitas penggunaan fitur-fitur ini, seperti yang terlihat dalam pengklasifikasian penyakit pada tanaman padi [8], [9], daun cendana [10], dan daun anggur [11].

Beberapa penelitian menunjukkan keunggulan algoritma MLP dalam klasifikasi citra. Syaputra A. memperoleh akurasi 97,4% pada data latih dalam klasifikasi daun tebu menggunakan MLP dengan VGG16 sebagai ekstraksi fitur [12], sedangkan Basri dkk. mencapai akurasi sebesar 93,33% pada identifikasi kalium pada tanaman kakao menggunakan MLP dengan kombinasi fitur LBP, GLCM, dan HSV [7]. Di sisi lain, penelitian yang dilakukan oleh She X. dkk. menunjukkan bahwa model DenseNet memiliki kinerja yang lebih unggul dibandingkan ResNet-50, VGG16, InceptionV3, dan Xception dalam mengklasifikasikan kondisi kulit [13].

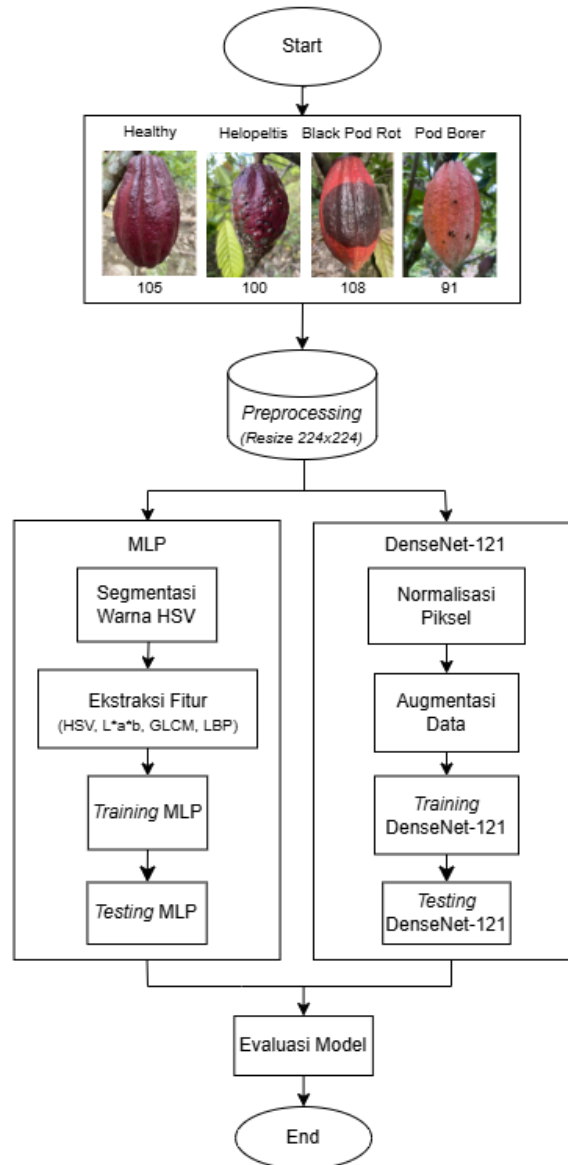
Berdasarkan penelitian sebelumnya, kombinasi fitur warna dan tekstur terbukti efektif membantu meningkatkan keakuratan dalam klasifikasi citra. Selain itu, algoritma MLP menunjukkan performa yang baik ketika dikombinasikan dengan fitur yang tepat, sedangkan model *deep learning* seperti densenet-121 mampu melakukan ekstraksi fitur secara otomatis dari citra mentah. Namun, sebagian besar penelitian sebelumnya hanya menggunakan satu pendekatan saja, baik yang berbasis ekstraksi fitur atau *deep learning*, tanpa membandingkan keduanya dengan dataset yang sama.

Oleh karena itu, kebaruan dari penelitian ini adalah membandingkan antara metode berbasis ekstraksi fitur dengan menggunakan MLP dan metode *deep learning* menggunakan densenet-121 dalam mengklasifikasikan empat kondisi buah kakao. Perbandingan dilakukan dengan menggunakan dataset yang sama, serta dievaluasi dengan metrik yang sama. Penelitian ini bertujuan untuk membandingkan kinerja algoritma MLP yang memanfaatkan penggunaan fitur warna (HSV dan L^*a^*b) serta fitur tekstur (GLCM dan LBP) dengan DenseNet-121 dalam mengklasifikasikan empat kondisi buah kakao. Diharapkan penelitian ini dalam memberikan pemahaman yang lebih mendalam mengenai keterbatasan masing-masing pendekatan, serta menentukan metode yang paling efektif dalam klasifikasi citra buah kakao.

2. METODE PENELITIAN

Penelitian ini bertujuan membandingkan kinerja algoritma *Multilayer Perceptron* (MLP) yang memanfaatkan fitur warna (HSV dan L^*a^*b) dan tekstur (GLCM dan LBP) dengan model DenseNet-121 dalam mengklasifikasikan empat kondisi buah kakao, yaitu: *healthy*, *black pod rot*, *pod borer*, dan *helopeltis*. Evaluasi dilakukan menggunakan metrik akurasi, presisi, *recall*, dan F1-score untuk mengetahui metode dengan performa terbaik. Seluruh implementasi dilakukan di Google Colab menggunakan Python 3.11. MLP diimplementasikan dengan framework PyTorch, sedangkan DenseNet-121 menggunakan TensorFlow/Keras, pada

lingkungan dengan spesifikasi RAM 4 GB dan prosesor Intel Core i3. Kedua model menggunakan skenario pembagian data yang sama, yaitu 70% data latih, 15% data validasi, dan 15% data uji. Tahapan penelitian yang dilakukan digambarkan pada Gambar 1 berikut:



Gambar 1. Tahapan klasifikasi

2.1. Dataset

Dataset yang digunakan merupakan data primer yang dikumpulkan langsung dari kebun kakao di Desa Riso, Kec. Tapango, Kab. Polewali Mandar, Sulawesi Barat dan Dusun Talongga, Desa Seppong, Kec. Tameroddo, Kab. Majene, Sulawesi Barat yang berjumlah 404 data. Data ini terdiri dari empat kategori yaitu: *Healthy* 105 citra, *Black pod rot* 108 citra, *Pod Borer* 91 citra dan *Helopeltis* 100. Pemilihan ketiga penyakit ini karena merupakan penyakit yang paling banyak ditemukan di Riso dan Talongga.

2.2. Preprocessing

Pada tahap *preprocessing*, ukuran semua citra diubah (*resize*) agar seragam menjadi 224×224 piksel. Tahap ini bertujuan untuk menyamakan dimensi input sebelum proses pembelajaran model, sehingga mempermudah proses komputasi, serta mengoptimalkan kinerja algoritma dalam melakukan klasifikasi, tanpa menghilangkan informasi penting dari citra.

2.3. Segmentasi Warna

Tahap segmentasi dilakukan untuk memisahkan objek dari latar belakang (*background*) sehingga mempermudah proses analisis [14]. Segmentasi warna adalah proses memisahkan bagian-bagian dalam sebuah citra berdasarkan warna yang ada dalam citra tersebut [15]. Segmentasi warna dilakukan melalui tiga kali masking menggunakan metode HSV dengan nilai *lower* dan *upper*, nilai ini merupakan nilai ambang batas untuk deteksi objek. Nilai *lower* dan *upper* yang digunakan sebagai berikut:

Tabel 1. Nilai *lower* dan *upper*

| Nilai | Masking 1 | Masking 2 | Masking 3 |
|-------|-----------------|----------------|-----------------|
| Lower | [120, 25, 30] | [0, 60, 35] | [105, 0, 0] |
| Upper | [179, 255, 255] | [35, 255, 255] | [179, 255, 150] |

Ketiga rentang nilai HSV pada table 1 ditentukan berdasarkan karakteristik warna buah kakao yang berbeda-beda. Masking 1 fokus pada warna merah keunguan, masking 2 fokus pada warna kuning kecoklatan, dan masking 3 fokus pada area gelap atau bagian buah yang coklat tua.

2.4. Ekstraksi Fitur

Ekstraksi fitur dilakukan untuk mendapatkan karakteristik dari citra dengan mengambil nilai fitur warna menggunakan HSV dan L^*a^*b dan fitur tekstur menggunakan GLCM dan LBP, tahapan ini dilakukan dengan memanfaatkan *library OpenCV* dan *scikit-image*.

a. HSV (*Hue, Saturation, Value*)

Fitur HSV terdiri dari tiga elemen penting, yaitu *Hue*, *Saturation*, dan *Value*. *Hue* merepresentasikan jenis warna (seperti merah, hijau, atau biru), dan juga merupakan karakteristik yang dimiliki oleh warna sehingga dapat dikenali. *Saturation* menggambarkan intensitas atau kemurnian warna, dengan rentang 0 (abu-abu) sampai 100% (warna sepenuhnya). Di sisi lain, *Value* menunjukkan seberapa terang suatu warna, dari 0 (hitam) hingga 100% (terang maksimal), dan digunakan untuk menentukan apakah warna tersebut tampak terang atau gelap [7], [16].

b. L^*a^*b

Ruang warna L^*a^*b (*Lightness, *a, *b*) adalah model tiga dimensi yang mewakili warna berdasarkan tingkat kecerahan (L^*), spektrum hijau/merah (a^*), dan spektrum biru/kuning (b^*). Komponen L^* menunjukkan tingkat kecerahan, a^* mewakili dominasi warna hijau atau merah, sedangkan b^* menunjukkan dominasi warna biru atau kuning. Model ini lebih mencerminkan persepsi manusia terhadap warna dengan memisahkan luminansi dan krominansi [17].

c. GLCM

Gray Level Co-occurrence Matrix (GLCM) adalah matriks yang menggambarkan hubungan antar piksel dengan menghitung frekuensi kemunculan pasangan piksel yang memiliki level keabuan tertentu pada jarak dan sudut tertentu. Metode ini digunakan untuk mengambil ciri tekstur dari sebuah citra dengan menganalisis keterkaitan spasial antara piksel-piksel yang ada [18].

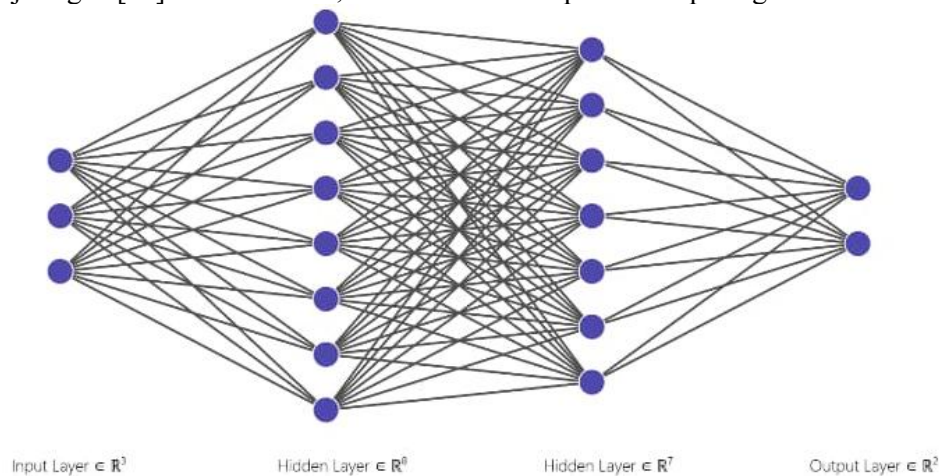
d. LBP

LBP mendeskripsikan pola lokal dalam gambar dengan membandingkan intensitas piksel pusat terhadap piksel-piksel di sekitarnya melalui dua langkah: *thresholding* dan *encoding*. Pada *thresholding*, piksel sekitar diberi nilai 1 jika intensitasnya \geq piksel pusat,

atau 0 jika lebih kecil, kemudian di-encode menjadi pola biner yang merepresentasikan tekstur lokal [19].

2.5. Training MLP

Multilayer Perceptron (MLP) merupakan salah satu jenis jaringan saraf tiruan *Artificial Neural Network* (ANN) yang dikembangkan dari konsep *perceptron*. MLP termasuk dalam arsitektur *feedforward* yang memiliki satu atau lebih lapisan tersembunyi (*hidden layer*). Jaringan ini umumnya terdiri atas lapisan masukan atau *input layer*, lapisan tersembunyi, dan lapisan keluaran (*output layer*), di mana setiap lapisan memiliki peran khusus. Lapisan masukan berfungsi menerima sinyal atau vektor dari luar dan meneruskannya ke seluruh *neuron* pada lapisan tersembunyi. Sementara itu, lapisan keluaran bertugas menerima sinyal hasil pemrosesan dari lapisan tersembunyi dan menghasilkan keluaran akhir berupa nilai atau kelas dari keseluruhan jaringan [20]. Secara visual, arsitektur MLP dapat dilihat pada gambar 2 berikut:



Gambar 2. Arsitektur MLP

Model ini dirancang untuk memproses fitur hasil ekstraksi warna (HSV, L^*a^*b) dan tekstur (GLCM, LBP). Arsitektur yang digunakan terdiri dari lapisan input yang sesuai dengan jumlah fitur, dua lapisan tersembunyi atau *hidden layer* masing-masing 128 dan 64 neuron dengan aktivasi ReLU dan *batch normalization*, serta lapisan output 4 neuron. Fungsi aktivasi *Softmax* tidak ditambahkan secara eksplisit karena telah terintegrasi dalam fungsi *CrossEntropyLoss*. Dataset dibagi menjadi data latih: uji: validasi dengan skenario 70:15:15 menggunakan *train_test_split* dari *scikit-learn* dan distandarisasi menggunakan *StandardScaler*. Pelatihan dilakukan dengan *batch size* 64, maksimum 50 *epoch*, optimizer Adam, dan *learning rate* 0,1-0,00001. *Early stopping* diterapkan dengan parameter *patience* sebanyak 10 *epoch*, untuk memantau *loss* pada data validasi. Pelatihan dihentikan lebih cepat jika tidak ada peningkatan, dan bobot model dikembalikan ke kondisi terbaiknya. Selain itu, ditambahkan regulasi L2 dengan $\lambda = 10^{-5}$ untuk mencegah *overfitting*.

2.6. Normalisasi Piksel

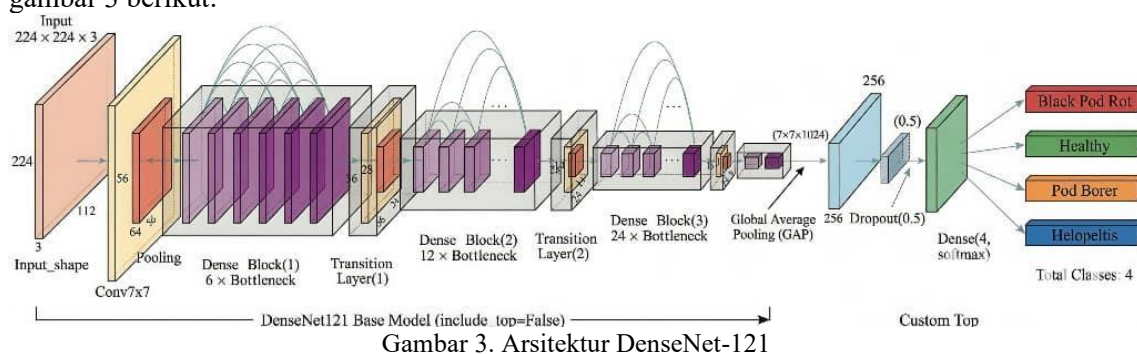
Pada model DenseNet-121, setiap piksel dari dari citra dinormalisasikan menggunakan fungsi *preprocess_input* yang ada di Keras. Proses ini penting dalam metode *transfer learning* karena menyelaraskan distribusi *input* sesuai dengan bobot yang sudah dilatih dari ImageNet. Sehingga, aliran gradien selama pelatihan menjadi lebih stabil dan fitur visual yang sudah dipelajari oleh model dasar dapat dimanfaatkan secara optimal. Normalisasi diterapkan secara konsisten pada data latih, validasi, maupun uji.

2.7. Augmentasi Data

Augmentasi data diterapkan pada data latih menggunakan *ImageDataGenerator* untuk mencegah *overfitting* serta meningkatkan kemampuan generalisasi model untuk penyakit kakao mensimulasikan variasi kondisi di lapangan seperti sudut pandang dan pencahayaan. Transformasi acak yang dilakukan meliputi rotasi 30°, pergeseran 20%, shear 20%, zoom 20%, dan pembalikan horizontal (*horizontal flip*) dengan metode pengisian piksel kosong dari nearest. Seluruh augmentasi diterapkan secara acak pada setiap *batch* selama proses pelatihan, data validasi dan uji tidak melalui melalui augmentasi agar penilaian kerja model mencerminkan kemampuannya pada data asli.

2.8. Training DenseNet-121

DenseNet-121 menerapkan koneksi yang padat (*dense connectivity*). Setiap lapisan di dalam satu blok menerima *input* dari semua lapisan sebelumnya, sehingga memaksimalkan aliran informasi dan efisiensi parameter. Arsitektur DenseNet-121 yang digunakan dapat dilihat pada gambar 3 berikut:



Gambar 3. Arsitektur DenseNet-121

Arsitektur DenseNet-121 yang digunakan seperti pada Gambar 3 adalah model yang telah dilatih sebelumnya pada ImageNet dengan bobot awal yang dimuat tanpa lapisan klasifikasi aslinya (*include_top=False*) dan ukuran input 224×224 piksel. Di atas model dasar, ditambahkan lapisan *GlobalAveragePooling2D*, lapisan dense dengan 256 unit yang menggunakan aktivasi ReLU, dropout 0,5, serta lapisan dense output dengan 4 unit menggunakan aktivasi softmax untuk klasifikasi empat kelas. Model dasar ini diatur agar tidak dapat dilatih, sehingga hanya lapisan tambahan yang akan diperbarui (*feature extraction*). Pelatihan dilakukan dengan menggunakan optimizer Adam dan *learning rate* 0,1-0,00001 dengan nilai maksimal *epoch* 25 dan 50. *Early stopping* memantau *loss* validasi dengan parameter *patience* sebanyak 8 *epoch* dan mengembalikan bobot terbaik. Pengujian dilakukan beberapa kali dengan menggunakan 3 *seed* (42, 43, dan 99). *Batch size* = 16.

2.9. Evaluasi Model

Pengujian model dalam mengklasifikasi kondisi buah kakao menjadi 4 kategori digunakan menggunakan metrik akurasi (*accuracy*) yang ditunjukkan pada persamaan berikut:

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN} \times 100\% \quad (1)$$

Keterangan:

TP = True Positif

TN = True Negatif

FP = False Positif

FN = False Negatif

2.10. Testing MLP

Pengujian MLP dilakukan menggunakan 64 sampel data uji yang diperoleh melalui pembagian data manual per kelas (*stratified manual split*) dengan rasio 70:15:15. Fitur data uji distandarisasi menggunakan StandardScaler yang sama dengan data latih untuk mencegah kebocoran data. Model dijalankan dalam mode evaluasi (*model.eval()*) tanpa perhitungan gradien (*torch.no_grad()*) dan menggunakan *data loader* tidak acak (*shuffle=False*) demi efisiensi komputasi. Seluruh proses diulang tiga kali menggunakan *random seed* berbeda (42, 43, dan 99) untuk memastikan validitas dan ketangguhan hasil terhadap variasi pembagian data.

2.11. Testing DenseNet-121

Pengujian DenseNet-121 dilakukan menggunakan 64 sampel data uji yang telah dinormalisasi dengan menggunakan *process input* tanpa augmentasi data. Model terbaik dimuat dari *checkpoint* penyimpanan bobot. Prediksi dilakukan menggunakan fungsi *model.predict()* dan generator data yang tidak diacak (*shuffle = False*). Hasil prediksi dievaluasi menggunakan metrik akurasi, presisi, recall, dan F1-Score. Sama seperti MLP, pengujian dilakukan tiga kali dengan *seed* acak yang berbeda agar hasilnya tetap konsisten.

3. HASIL DAN PEMBAHASAN

Pengujian dilakukan menggunakan dua model yang berbeda, yaitu MLP dengan 15 kombinasi fitur (HSV, L*a*b, GLCM, LBP) dan DenseNet-121. Dataset dibagi dengan rasio 70:15:15 (latih, validasi, uji). Seluruh pengujian dilakukan menggunakan *optimizer* Adam dan *early stopping* untuk mencegah *overfitting*.

3.1. Hasil Pengujian Model MLP

Untuk mengevaluasi model MLP, dilakukan uji coba dengan menggunakan 15 kombinasi fitur yang berbeda. Setiap skenario kombinasi fitur diuji dengan menggunakan 3 nilai *random seed* yang berbeda yaitu 42, 43, dan 99 untuk memastikan validitas hasil dan menghindari bias pada pembagian data. Tiap kombinasi juga dijalankan dengan nilai *learning rate* yang berbeda (0,1 hingga 0,00001). Tabel menampilkan hasil pengujian pada *learning rate* terbaik untuk setiap kombinasi fitur dan *seed*.

Tabel 2. Performa model MLP (*learning rate* terbaik)

| Fitur | LR Terbaik | Seed | Akurasi (%) |
|----------|------------|------|-------------|
| LAB | 0,001 | 42 | 53,12 |
| LAB | 0,001 | 43 | 67,19 |
| LAB | 0,001 | 99 | 75,00 |
| HSV | 0,0001 | 42 | 37,50 |
| HSV | 0,0001 | 43 | 57,81 |
| HSV | 0,0001 | 99 | 51,56 |
| LBP | 0,00001 | 42 | 45,31 |
| LBP | 0,00001 | 43 | 56,25 |
| LBP | 0,00001 | 99 | 50,00 |
| GLCM | 0,001 | 42 | 48,44 |
| GLCM | 0,001 | 43 | 67,19 |
| GLCM | 0,001 | 99 | 54,69 |
| GLCM+LAB | 0,001 | 42 | 59,38 |
| GLCM+LAB | 0,001 | 43 | 78,12 |
| GLCM+LAB | 0,001 | 99 | 78,12 |
| GLCM+LBP | 0,001 | 42 | 43,75 |
| GLCM+LBP | 0,001 | 43 | 59,38 |
| GLCM+LBP | 0,001 | 99 | 45,31 |
| HSV+GLCM | 0,0001 | 42 | 45,31 |
| HSV+GLCM | 0,0001 | 43 | 67,19 |

| | | | |
|---------------------|--------------|-----------|--------------|
| HSV+GLCM | 0,0001 | 99 | 59,38 |
| HSV+LAB | 0,001 | 42 | 57,81 |
| HSV+LAB | 0,001 | 43 | 64,06 |
| HSV+LAB | 0,001 | 99 | 75,00 |
| HSV+LBP | 0,1 | 42 | 59,38 |
| HSV+LBP | 0,1 | 43 | 43,75 |
| HSV+LBP | 0,1 | 99 | 42,19 |
| LBP+LAB | 0,0001 | 42 | 45,31 |
| LBP+LAB | 0,0001 | 43 | 67,19 |
| LBP+LAB | 0,0001 | 99 | 56,25 |
| HSV+GLCM+LAB | 0,001 | 42 | 65,62 |
| HSV+GLCM+LAB | 0,001 | 43 | 64,06 |
| HSV+GLCM+LAB | 0,001 | 99 | 79,69 |
| HSV+GLCM+LBP | 0,0001 | 42 | 43,75 |
| HSV+GLCM+LBP | 0,0001 | 43 | 67,15 |
| HSV+GLCM+LBP | 0,0001 | 99 | 48,44 |
| GLCM+LAB+LBP | 0,001 | 42 | 59,38 |
| GLCM+LAB+LBP | 0,001 | 43 | 62,50 |
| GLCM+LAB+LBP | 0,001 | 99 | 53,12 |
| HSV+LAB+LBP | 0,0001 | 42 | 50,00 |
| HSV+LAB+LBP | 0,0001 | 43 | 64,06 |
| HSV+LAB+LBP | 0,0001 | 99 | 56,25 |
| HSV+LAB+GLCM+LBP | 0,0001 | 42 | 51,56 |
| HSV+LAB+GLCM+LBP | 0,0001 | 43 | 67,19 |
| HSV+LAB+GLCM+LBP | 0,0001 | 99 | 56,25 |

Berdasarkan Tabel 2, hasil pengujian menunjukkan jika penggunaan kombinasi fitur secara konsisten memberikan performa yang lebih unggul dalam mengklasifikasikan data dibanding hanya menggunakan satu fitur saja. Kombinasi HSV+GLCM+L*a*b pada *seed* 99 memberikan akurasi tertinggi sebesar 79,69% dengan *learning rate* 0,001. Kelebihan kombinasi HSV+GLCM+L*a*b terletak pada kerja sama antar ketiga fitur: HSV dan L*a*b bersama-sama mengambil informasi warna yang *robust* terhadap variasi pencahayaan, sedangkan GLCM melengkapi dengan ekstraksi pola tekstur permukaan. Integrasi ketiga fitur ini memungkinkan model membedakan penyakit yang memiliki kemiripan warna namun berbeda tekstur (atau sebaliknya) secara lebih akurat dibandingkan penggunaan fitur tunggal atau kombinasi parsial.

Model yang hanya menggunakan satu fitur justru menunjukkan hasil yang lebih buruk dan tidak stabil. Akurasi terendah tercatat pada fitur HSV (*seed* 42) sebesar **37,50%**, sementara fitur tunggal terbaik yaitu L*a*b hanya mencapai **75,00%** pada *seed* 99. Rendahnya performa fitur tunggal mengindikasikan bahwa satu jenis deskriptor saja tidak cukup untuk menangkap kompleksitas visual dari variasi gejala penyakit seperti *black pod rot*, *helopeltis*, dan *pod borer* yang memiliki kemiripan pola.

Selain itu, terdapat variasi akurasi yang signifikan antar *seed* pada konfigurasi fitur yang sama. Sebagai contoh, pada kombinasi **GLCM+L*a*b**, akurasi berfluktuasi dari 59,38% (*seed* 42) hingga 78,12% (*seed* 43 dan 99). Variasi serupa juga terlihat pada hampir semua skenario pengujian. Hal ini menegaskan sensitivitas model MLP terhadap pembagian data acak, terutama mengingat ukuran dataset yang terbatas. Oleh karena itu, penggunaan tiga *random seed* berbeda (42, 43, dan 99) dalam evaluasi ini menjadi krusial untuk memastikan bahwa laporan kinerja model bersifat objektif, *robust*, dan tidak bias akibat keberuntungan dalam pemisahan data (*lucky split*). Secara keseluruhan, meskipun kombinasi fitur berhasil meningkatkan akurasi hingga mendekati 80%, hasil ini masih menunjukkan celah kinerja jika dibandingkan dengan pendekatan *deep learning*.

3.2. Hasil Pengujian Model DenseNet-121

Untuk memastikan hasil yang valid dan menghindari bias dalam membagi data (*lucky split*), setiap skenario pengujian diulang tiga kali dengan menggunakan *random seed* yang berbeda yaitu 42, 43, dan 99. Model diuji pada dua variasi jumlah *epoch* yaitu 25 dan 50 serta lima variasi *learnin rate* dari 0,1 hingga 0,00001. Tabel 3 menampilkan hasil pengujian pada model DenseNet-121 untuk setiap kombinasi jumlah *epoch*, *learning rate*, dan nilai *seed*.

Tabel 3. Hasil pengujian *DenseNet-121*

| Epoch | LR | Seed | Akurasi (%) |
|-----------|--------------|-----------|--------------|
| 25 | 0,1 | 42 | 37,50 |
| 25 | 0,1 | 43 | 43,75 |
| 25 | 0,1 | 99 | 28,12 |
| 25 | 0,01 | 42 | 84,38 |
| 25 | 0,01 | 43 | 90,62 |
| 25 | 0,01 | 99 | 85,54 |
| 25 | 0,001 | 42 | 78,12 |
| 25 | 0,001 | 43 | 92,19 |
| 25 | 0,001 | 99 | 82,81 |
| 25 | 0,0001 | 42 | 76,50 |
| 25 | 0,0001 | 43 | 89,06 |
| 25 | 0,0001 | 99 | 82,81 |
| 25 | 0,00001 | 42 | 46,88 |
| 25 | 0,00001 | 43 | 50,00 |
| 25 | 0,00001 | 99 | 54,69 |
| 50 | 0,1 | 42 | 37,50 |
| 50 | 0,1 | 43 | 43,75 |
| 50 | 0,1 | 99 | 28,12 |
| 50 | 0,01 | 42 | 84,38 |
| 50 | 0,01 | 43 | 90,62 |
| 50 | 0,01 | 99 | 84,38 |
| 50 | 0,001 | 42 | 78,12 |
| 50 | 0,001 | 43 | 92,19 |
| 50 | 0,001 | 99 | 82,81 |
| 50 | 0,0001 | 42 | 76,56 |
| 50 | 0,0001 | 43 | 89,06 |
| 50 | 0,0001 | 99 | 82,81 |
| 50 | 0,00001 | 42 | 65,62 |
| 50 | 0,00001 | 43 | 71,88 |
| 50 | 0,00001 | 99 | 68,75 |

Berdasarkan Tabel 3, DenseNet-121 mencapai kinerja optimal pada *learning rate* 0,001 dan *seed* 43, dengan mencapai akurasi tertinggi sebesar 92,19% pada *epoch* 25 maupun 50. Konfigurasi dengan *learning rate* 0,01 juga menunjukkan hasil yang konsisten baik dengan akurasi berkisar 78,12% hingga 92,19%. Hasil ini menunjukkan bahwa pada *learning rate* tersebut, model mampu memperbarui bobot secara efektif tanpa membuat model menjadi tidak stabil. Namun, *learning rate* yang terlalu besar seperti 0,1 menyebabkan model terus menyimpang, sehingga akurasinya menjadi sangat rendah, antara 28,12% hingga 43,75%. Hal ini terjadi karena pembaruan bobot yang terlalu cepat yang menyebabkan model gagal menemukan titik minimum *loss*. Di sisi lain, *learning rate* yang terlalu kecil (0,00001) juga menyebabkan akurasi yang rendah (46,88% hingga 71,88%) hal ini karena perubahan bobot yang terjadi secara perlahan, sehingga tidak mampu mencapai tingkat konvergensi optimal dalam jumlah *epoch* yang telah ditentukan.

Hasil angatara 25 dan 50 *epoch* menunjukkan hasil yang hampir sama pada sebagian konfigurasi. Ini menunjukkan bahwa model sudah stabil pada *epoch* 25 dan menambahkan *epoch*

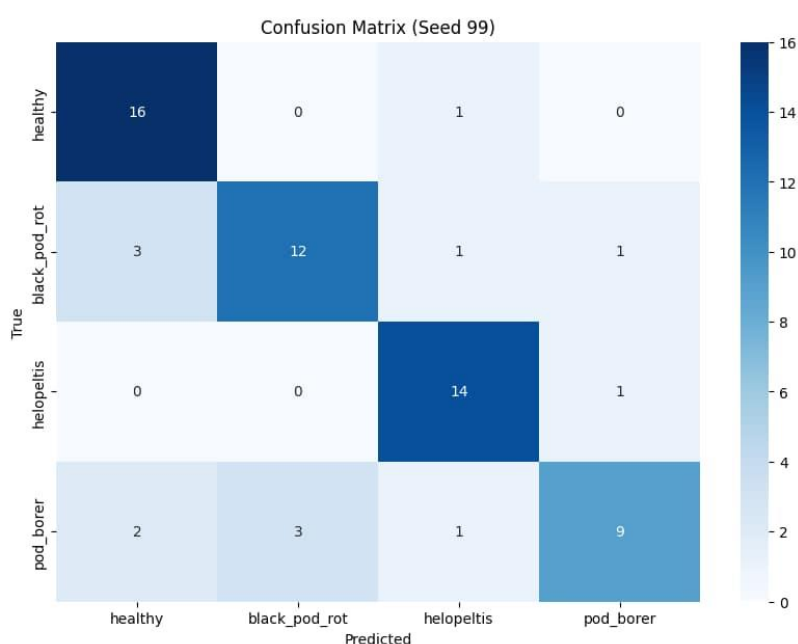
tambahan tidak memberikan peningkatan yang berarti. Secara keseluruhan, variasi akurasi antar *seed* pada konfigurasi terbaik tidak terlalu besar, menunjukkan bahwa model DenseNet-121 cukup stabil dan hasilnya dapat dipercaya.

3.3. Analisis Kesalahan Klasifikasi (Confusion Matrix)

Untuk memahami lebih jauh mengenai tentang kemampuan model dalam mengenali setiap kelas dilakukan analisis menggunakan *confusion matrix* pada hasil pengujian terbaik dari kedua arsitektur dan *seed* dengan akurasi tertinggi. Gambar 4 dan 5 menunjukkan hasil prediksi kedua model untuk empat kondisi buah kakao: *healthy*, *black pod rot*, *pod borer* dan *helopeltis*.

a. Analisis Kesalahan Klasifikasi Model MLP

Gambar 4 menyajikan *confusion matrix* untuk model MLP dengan kombinasi fitur terbaik (HSV+GLCM+L*a*b) pada *seed* dengan akurasi tertinggi yaitu 99. Analisis ini penting untuk mengetahui kelas mana yang paling tepat diprediksi dan kelas mana yang sering terjadi kesalahan dalam pengklasifikasian.



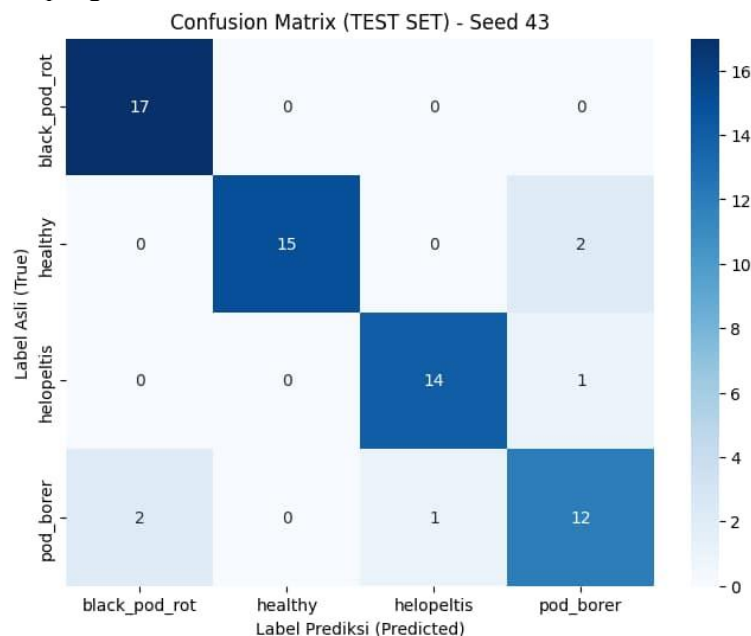
Gambar 4. *Confusion matrix* akurasi tertinggi pengujian mlp

Gambar 4 menampilkan *confusion matrix* dari model MLP dengan kombinasi fitur terbaik yaitu HSV+GLCM+L*a*b pada pengujian *seed* 99. Nilai pada diagonal utama menunjukkan berapa banyak prediksi yang benar (*True Positive*) untuk setiap kelas. Model menunjukkan hasil yang sangat baik pada kategori *healthy*, mampu mengklasifikasikan 16 dari 17 citra secara tepat. Namun, terdapat 1 citra sehat yang salah diprediksi sebagai *helopeltis*, kemungkinan akibat adanya bercak alami atau variasi warna pada kulit buah yang menyerupai gejala awal penyakit.

Pada kelas penyakit, model berhasil memprediksi 12 dari 17 citra *black pod rot* dengan tepat. Terdapat 3 citra yang salah dikategorikan sebagai *healthy*, hal ini menunjukkan bahwa pada beberapa kasus infeksi awal, perubahan warna dan tekstur belum cukup jelas untuk terdeteksi oleh metode ekstraksi fitur manual. Kesalahan klasifikasi terberat terjadi pada kelas *pod borer*, hanya 9 dari 15 citra yang berhasil diprediksi dengan benar. Tiga citra terprediksi sebagai *black pod rot* dan dua citra menjadi *healthy*. Hal ini menunjukkan bahwa meskipun gabungan tiga fitur telah meningkatkan kinerja model secara keseluruhan, model masih kesulitan membedakan visual antara lubang gerakan dengan dengan area busuk gelap. Kedua kondisi tersebut memiliki warna dan tekstur yang cenderung mirip.

b. Analisis Kesalahan Klasifikasi Model Densenet-121

Gambar 5 menunjukkan hasil prediksi model DenseNet-121 pada uji coba terbaik (Iterasi 2, Seed 43), yang berhasil mencapai akurasi tertinggi sebesar 90,62%. Analisis ini penting untuk mengetahui kelas mana yang paling tepat diprediksi dan kelas mana yang sering terjadi kesalahan dalam pengklasifikasian.



Gambar 5. Confusion matrix akurasi tertinggi pengujian densenet-121

Gambar 5 menampilkan *confusion matrix* dari model DenseNet-121 pada pengujian seed 43 yang menghasilkan akurasi tertinggi sebesar 90,62%. Nilai pada diagonal utama merepresentasikan jumlah prediksi yang benar (*true positive*) untuk setiap kelas. Model berhasil mengklasifikasikan seluruh citra kelas *black pod rot* dengan sempurna (17 dari 17 citra). Untuk kelas *healthy*, model memprediksi dengan benar 15 citra, namun terdapat 2 citra yang salah diklasifikasikan sebagai *pod borer*. Pada kelas *helopeltis*, 14 citra terprediksi benar dengan 1 kesalahan ke kelas *pod borer*. Sementara itu, kelas *pod borer* memiliki 12 prediksi benar, dengan 2 citra tertukar ke *black pod rot* dan 1 citra ke *helopeltis*.

Kesalahan klasifikasi yang terjadi sebagian besar disebabkan oleh kemiripan karakteristik visual antar kelas penyakit. Misalnya, kesalahan pada kelas *healthy* yang terprediksi sebagai *pod borer* kemungkinan akibat adanya bercak alami atau bayangan pada buah sehat yang menyerupai lubang gerakan kecil. Demikian pula, tertukarnya *pod borer* dengan *black pod rot* disebabkan oleh kesamaan fitur warna gelap pada area penyakit.

4. KESIMPULAN

Penelitian ini menunjukkan bahwa metode *Deep Learning* menggunakan DenseNet-121 lebih efektif dibandingkan metode MLP yang menggunakan fitur manual untuk klasifikasi empat kondisi buah kakao yaitu *healthy*, *black pod rot*, *pod borer*, dan *helopeltis*, dengan akurasi tertinggi 92,19% pada *learning rate* 0,001 (*seed* 43), sedangkan metode MLP terbaik hanya mencapai akurasi 79,69% dengan menggunakan kombinasi fitur HSV, GLCM dan L^*a^*b pada *learning rate* 0,001 dengan *seed* 99. DenseNet lebih unggul karena kemampuannya dalam mengekstraksi fitur secara otomatis dari tingkat rendah hingga tinggi melalui lapisan konvolusi bertumpuk serta memanfaatkan bobot *pre-trained* ImageNet untuk generalisasi yang lebih baik pada dataset kecil, sedangkan MLP terbatas oleh kualitas fitur manual yang kehilangan konteks

spasial. Penggunaan *learning rate* sangat memengaruhi performa kedua model, nilai yang terlalu besar dapat menyebabkan divergensi atau kondisi dimana model gagal belajar dan *error* makin besar dan nilai terlalu kecil memperlambat konvergensi (proses model mencapai akurasi optimal), namun penerapan *multiple seeds* (42, 43, 99) membuktikan hasil eksperimen stabil dan tidak bergantung pada kebetulan pembagian data. Untuk penelitian selanjutnya, disarankan untuk menerapkan *fine-tuning* pada DenseNet, mengeksplorasi arsitektur *deep learning* lain seperti ResNet atau EfficientNet, serta melakukan augmentasi data yang lebih variatif dan optimasi *hyperparameter* MLP secara sistematis untuk meningkatkan *robustness* model.

DAFTAR PUSTAKA

- [1] I. Komang *et al.*, “Identifikasi Penyakit Buah Kakao Jembrana Menggunakan Metode RESNET152V2,” 2024. [Online]. Available: www.jembranakab.go.id
- [2] S. Putra, Y. Ferry, and R. Harni, “Pengendalian penyakit busuk buah kakao menggunakan Trichoderma dan pupuk Kalium,” *Jurnal Kultivasi*, vol. 21, no. 2, Aug. 2022, doi: 10.24198/kultivasi.v21i2.36807.
- [3] G. A. S. Gado and P. N. Primandari, “Sistem Klasifikasi Berbasis Android untuk Penyakit Buah Kakao Menggunakan CNN,” *Jurnal Teknologi Terpadu*, vol. 11, pp. 27–35, 2025.
- [4] R. Robet, J. T. K. Perangin Angin, and T. H. Siregar, “Attention Augmented Deep Learning Model for Enhanced Feature Extraction in Cacao Disease Recognition,” *Sinkron : Jurnal dan Penelitian Teknik Informatika*, vol. 9, no. 4, pp. 1965–1977, Oct. 2025, doi: 10.33395/sinkron.v9i4.15249.
- [5] D. Mamadou, K. J. Ayikpa, A. B. Ballo, and B. M. Kouassi, “Cocoa Pods Diseases Detection by MobileNet Confluence and Classification Algorithms,” *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 9, pp. 344–352, 2023, doi: 10.14569/IJACSA.2023.0140937.
- [6] K. R. Ariawan, A. A. G. Ekayana, I. P. Y. Indrawan, K. R. Winatha, and I. N. A. F. Setiawan, “Performance Comparasion of DenseNet-121 and MobileNetV2 for Cacao Fruit Disease Image Classification,” *Indonesian Journal of Data and Science*, vol. 6, no. 1, pp. 30–38, Mar. 2025, doi: 10.56705/ijodas.v6i1.233.
- [7] Basri, H. A. Karim, M. Assidiq, M. Arafah, and F. Rahmadani, “Multilayer Perceptron Model with Feature Extraction for Potassium Deficiency Identification of Cocoa Plants,” *International Journal on Informatics Visualization*, vol. 9, no. 1, pp. 324–332, 2025, [Online]. Available: www.joiv.org/index.php/joiv
- [8] Siti Saniah and Mhd. Furqan, “Classification Of Rice Plant Diseases Using K-Nearest Neighbor Algorithm Based On Hue Saturation Value Color Extraction And Gray Level Co-Occurrence Matrix Features,” *JURNAL TEKNOLOGI DAN OPEN SOURCE*, vol. 7, no. 2, pp. 212–223, Dec. 2024, doi: 10.36378/jtos.v7i2.3972.
- [9] R. Jordy and D. Ariatmanto, “Perbandingan Metode Ekstraksi Fitur LBP, GLCM, dan Canny dalam Klasifikasi Penyakit Daun Padi dengan KNN,” *Jurnal Bangkit Indonesia*, vol. 14, no. 2, pp. 44–51, Oct. 2025, doi: 10.52771/bangkitindonesia.v14i2.452.
- [10] K. Jumarto *et al.*, “Identifikasi penyakit pada foliage tanaman cendana menggunakan algoritma ID3 berdasarkan fitur GLCM dan Color Moment,” *AITI: Jurnal Teknologi Informasi*, vol. 22, no. Maret, pp. 73–83, 2025.
- [11] F. Mahrus Fathoni, C. Aji Putra, A. Lina Nurlaili, F. Ilmu Komputer, U. Pembangunan Nasional, and J. Timur, “Klasifikasi Penyakit Daun Anggur menggunakan Metode K-Nearest Neighbor berdasarkan Gray Level Co-Occurrence Matrix,” *Jurnal Ilmiah Informatika dan Komputer*, vol. 3, no. 1, pp. 8–15, 2024, [Online]. Available: <https://ojs.unsiq.ac.id/index.php/biner>
- [12] A. Syaputra, “Klasifikasi Penyakit Daun pada Tebu dengan Pendekatan Algoritma K-Nearest Neighbors, Multilayer Perceptron dan Support Vector Machine,” *Jurnal Informatika Global*, vol. 15, no. 3, pp. 131–138, 2024.

- [13] X. She, H. Lu, Q. Liu, P. Xie, and Q. Xia, “Dermatological infrared thermal imaging with human-machine interaction image diagnostics interface using DenseNet,” *J. Radiat. Res. Appl. Sci.*, vol. 17, no. 1, p. 100826, Mar. 2024, doi: 10.1016/j.jrras.2024.100826.
- [14] R. T. Wahyuningrum, M. Jannah, B. D. Satoto, A. K. Sari, and A. D. Sensusiaty, “Segmentasi Citra X-Ray Dada Menggunakan Metode Modifikasi Deeplabv3+,” *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 10, no. 3, pp. 687–698, Jul. 2023, doi: 10.25126/jtiik.2023106754.
- [15] M. Utami and E. D. Putra, “Deteksi Objek Kualitas Daun Sawi Menggunakan Metode HSV Color dan Color Blob,” *JUSIBI (Jurna Sistem Informasi dan E-Bisnis)*, vol. 5, no. 2, pp. 85–93, 2023, [Online]. Available: <https://jurnal.ikhafi.or.id/index.php/jusibi>
- [16] Basri, Indrabayu, A. Achmad, and I. S. Areni, “Comparison of Image Extraction Model for Cocoa Disease Fruits Attack in Support Vector Machine Classification,” in *Proceedings - IEIT 2022: 2022 International Conference on Electrical and Information Technology*, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 46–51. doi: 10.1109/IEIT56384.2022.9967910.
- [17] P. F. Johari, N. Arifin, M. Muzaki, and M. S. A. Utama, “Corn Leaf Diseases Classification Using CNN with GLCM, HSV, and L*a*b* Features,” *Jurnal Teknik Informatika (Jutif)*, vol. 6, no. 2, pp. 709–722, Apr. 2025, doi: 10.52436/1.jutif.2025.6.2.4345.
- [18] M. Juventus, D. Deke, T. A. Anastasya, A. Diani, P. Saka, and E. Y. Puspaningrum, “Analisis Pengaruh Metode Ekstraksi Fitur Citra Batik terhadap Kinerja Klasifikasi SVM,” *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 9, no. 5, 2025, [Online]. Available: <https://www.kaggle.com/datasets/geryxg/cora>
- [19] R. A. Nindiaputra, Y. V. Via, and H. Maulana, “Implementasi KNN dengan Fitur Ekstraksi Haralick dan Local Binary Pattern untuk Klasifikasi Penyakit Pneumonia pada Citra X-RAY,” *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 9, no. 6, pp. 10570–10578, 2025.
- [20] E. Fitri and D. Riana, “Analisa Perbandingan Model Prediction dalam Prediksi Harga Saham menggunakan Metode Linear Regression, Random Forest Regression dan Multilayer Perceptron,” *METHOMIKA Jurnal Manajemen Informatika dan Komputerisasi Akuntansi*, vol. 6, no. 1, pp. 69–78, Apr. 2022, doi: 10.46880/jmika.Vol6No1.pp69-78.