

IMPLEMENTASI TRANSFER *LEARNING* DENGAN PERBANDINGAN NILAI *LEARNING RATE* PADA KLASIFIKASI CITRA PENYAKIT DAUN TEBU BERBASIS WEB

Syafri Firmansyah, Eva Yulia Puspaningrum, Eka Prakarsa Mandyartha

Informatika, Universitas Pembangunan Nasional “Veteran” Jawa Timur

Jalan Raya Rungkut Madya No. 1 Surabaya, Indonesia

syafrifirmansyah@gmail.com

ABSTRAK

Pengolahan citra digital berfokus pada penciptaan dan analisis citra untuk mendapatkan informasi yang bermanfaat, termasuk dalam bidang pertanian. Kemajuan visi komputer membuka peluang untuk meningkatkan perlindungan tanaman, khususnya pada industri gula di Indonesia yang berpotensi menjadi produsen global. Namun, rendahnya produktivitas tebu disebabkan oleh perlakuan yang kurang intensif dan serangan penyakit yang mengurangi hasil panen. Penyakit pada daun tebu dapat menyebabkan kerugian ekonomi besar, sehingga deteksi dini dan penanganan yang tepat sangat penting untuk meningkatkan hasil produksi gula. Penelitian ini membandingkan beberapa arsitektur pretrained dalam teknik transfer learning untuk klasifikasi citra penyakit daun tebu, dengan fokus pada pengaruh variasi nilai learning rate. Data yang digunakan terdiri dari lima kelas kondisi daun tebu (sehat, Red Rot, Red Stripe, Rust, dan Bacterial Blight) dengan 500 gambar dari Kaggle. Lima arsitektur yang diuji meliputi AlexNet, DenseNet121, EfficientNet-B0, MobileNetV2, dan ResNet34, dengan dua nilai learning rate (0.0001 dan 0.00001). Evaluasi menggunakan metrik akurasi, precision, recall, dan F1-score menunjukkan DenseNet121 dengan learning rate 0.00001 memberikan akurasi terbaik (100%). Penelitian ini menyoroti pentingnya pemilihan arsitektur dan learning rate untuk klasifikasi citra penyakit daun tebu dan menunjukkan aplikasi web berbasis Flask sebagai alat deteksi dini yang efektif.

Kata Kunci: *Transfer Learning, Klasifikasi Citra, Penyakit Daun Tebu, Deep Learning, Learning Rate, DenseNet121, Flask.*

1. PENDAHULUAN

Pengolahan gambar digital mempelajari bagaimana gambar terbentuk, dikelola, dan dianalisis untuk menggali informasi yang bermanfaat bagi manusia [1]. Dengan kemajuan dalam bidang visi komputer, ada kesempatan untuk meningkatkan dan memperluas praktik perlindungan tanaman yang lebih baik dan untuk memperluas pasar visi komputer di bidang pertanian [2].

Indonesia memiliki potensi untuk menjadi produsen gula terbesar di dunia [3]. Selain itu, pasar gula Indonesia memiliki prospek yang menjanjikan dengan konsumsi 4,2-4,7 juta ton per tahun. Di Indonesia, tebu ditanam dalam skala pertanian dan perkebunan dengan tingkat pemeliharaan yang kurang intensif, sehingga produktivitas tebu di Indonesia rendah [4]. Penyakit tebu juga dapat menurunkan produktivitas. Karena mengurangi jumlah panen, penyebaran penyakit tebu memiliki dampak penurunan ekonomi yang signifikan. Untuk mengurangi biaya penggunaan pestisida dan pencemaran lingkungan yang disebabkan oleh obat penyakit, penanganan penyakit tebu juga memerlukan penemuan dan penanganan segera [5]. Permasalahan saat ini adalah bagaimana cara menentukan ciri dan proses klasifikasi dari jenis citra daun tebu, karena pada tanaman tebu tanda-tanda penyakit tersebut ditunjukkan pada daun yaitu berupa lesi, noda atau bercak. Pengelompokan jenis penyakit pada citra daun tebu merupakan salah satu hal yang penting untuk membantu meningkatkan kualitas produksi gula,

selain itu identifikasi jenis penyakit tebu lebih awal bisa membantu menentukan penanganan lebih awal juga. Hal ini dikarenakan noda-noda pada daun tebu menghambat proses fotosintesis yang dapat mengakibatkan menurunnya produksi gula, adapun jenis penyakit noda yang paling umum di Indonesia adalah noda cincin, noda karat, dan noda mosaik.[6].

Salah satu jenis neural network yang dapat mengenali objek dalam gambar adalah convolutional neural network (CNN). CNN adalah bagian dari Deep Neural Network, yang dikembangkan dari Multiplayer Perceptron (MLP), dan dirancang untuk mengolah data dua dimensi. Penelitian yang menggunakan metode CNN menunjukkan hasil yang sangat akurat. Saat ini, ada metode yang dapat digunakan untuk menggunakan model CNN yang sudah jadi untuk diterapkan pada dataset yang lebih kecil dengan hanya sedikit modifikasi pada layer klasifikasi. Metode ini dikenal sebagai pembelajaran transfer. Studi ini mencoba menggunakan transfer learning untuk mengklasifikasikan jenis-jenis ordo tumbuhan dengan dataset citra daun tumbuhan dan menganalisis perbedaan saat menggunakan model seperti VGG16, Inception-V3, dan ResNet50. Teknik transfer learning menggunakan model yang sudah dilatih sebelumnya (pretrained model) untuk mengklasifikasikan dataset yang baru sehingga tidak perlu melakukan transformasi lagi. CNN biasanya menggunakan berbagai lapisan untuk mengekstraksi dataset[8].

2. TINJAUAN PUSTAKA

Penelitian Faisal Mashuri dan Ultach Enri yang berjudul “Implementasi Transfer Learning Dalam Mendeteksi Penyakit Pada Daun Gandum”. Hasilnya, Inception V3 memberikan hasil yang lebih baik dibandingkan model yang lain dengan waktu komputasi yang rendah yaitu hanya 976 detik atau setara dengan 16 menit serta memiliki akurasi yang sangat tinggi [9].

Penelitian selanjutnya oleh Armin Lawi et al, yang berjudul “Implementasi Transfer Learning dan Multi-Channel CNN Pada Penyakit Daun Padi” menunjukkan bahwa data citra daun padi dengan total 120 citra dilakukan implementasi arsitektur multi Channel CNN dan menunjukkan hasil yang sangat baik yaitu dengan akurasi sebesar 95,83% pada masing masing two-channel dan three-channel CNN. Dan dapat disimpulkan bahwa arsitektur pembelajaran transfer multi-channel CNN lebih baik daripada arsitektur pembelajaran transfer sederhana dengan menggunakan pretrained network Xception, InceptionV3, dan VGG16 [1].

2.1. Penyakit Daun Tebu

Penyakit pada tanaman tebu disebabkan oleh berbagai pathogen. Salah satu penyakit umum adalah penyakit noda yang ditandai dengan infeksi berupa lesi atau bercak pada daun, yang dapat menghambat proses fotosintesis yang sangat penting bagi tanaman [6]. Infeksi ini dapat merusak daun secara keseluruhan, sehingga mengurangi produktivitas tebu [4].

2.2. Penyakit Noda pada Daun Tebu

Penyakit ini disebabkan oleh jamur yang menyerang daun tebu dan menghasilkan infeksi berupa lesi atau bercak. Lesi yang muncul dapat merusak daun, menghambat fotosintesis, dan menurunkan produksi tebu.

2.3. Penyakit Pokahbung

Disebabkan oleh jamur patogen *Fusarium moniliformae*, penyakit ini memiliki tiga tingkatan gejala: klorosis pada daun, garis merah kecoklatan pada batang, dan pembengkakan batang yang menyebabkan matinya tanaman.

2.4. Penyakit Rust Spot

Disebabkan oleh infeksi karat yang menyerang daun tebu, gejalanya berupa bercak kuning yang melebar menjadi coklat dan berukuran antara 2-10 mm, dengan beberapa kasus mencapai 30 mm.

2.5. Penyakit Blendok

Penyakit yang disebabkan oleh bakteri *Xanthomonas albineans*. Gejalanya berupa garis khlorosis pada daun yang berkembang menjadi tunas samping khlorosis, yang jika parah, menyebabkan tanaman mati.

2.6. Penyakit Mosaik

Disebabkan oleh beberapa virus seperti *Sugarcane Mosaic Virus* (SCMV) dan *Sugarcane Streak Mosaic Virus* (SCSMV), penyakit ini menimbulkan garis-garis dan noda berwarna hijau muda hingga kuning pada daun tanaman tebu.

2.7. Pengolahan Citra

Pengolahan citra adalah teknik yang digunakan untuk menganalisis gambar guna memperoleh informasi lebih spesifik tentang objek yang teramati. Setiap elemen gambar (pixel) memiliki setidaknya dua informasi yang menggambarkan posisinya serta warna. Dalam konteks pertanian, pengolahan citra dapat digunakan untuk mendeteksi penyakit tanaman dengan menganalisis citra daun, yang memberikan informasi penting untuk identifikasi penyakit pada tanaman tebu [7].

2.8. Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) adalah salah satu model pembelajaran dalam jaringan syaraf tiruan yang khusus dirancang untuk mengenali objek dalam citra. Dengan kemampuan untuk mengolah data dua dimensi, CNN telah terbukti efektif dalam klasifikasi citra dengan akurasi yang tinggi. Arsitektur CNN terdiri dari beberapa lapisan, termasuk konvolusi, pooling, dan fully connected layers, yang bekerja sama untuk mengekstraksi fitur dari citra [8]. Penelitian menunjukkan bahwa metode ini dapat dijadikan pendekatan yang tepat untuk klasifikasi citra penyakit daun tebu, dan teknik seperti transfer learning dapat meningkatkan performa model pada dataset yang lebih kecil [10].

2.9. Transfer Learning

Transfer Learning merupakan salah satu strategi untuk mempercepat proses pelatihan model dengan memanfaatkan model yang telah dilatih dengan dataset yang lebih besar. Teknik ini sangat bermanfaat ketika menerapkan model CNN pada dataset spesifik seperti citra daun tebu. Metode ini melakukan modifikasi pada arsitektur yang sudah ada untuk menyesuaikan dengan kebutuhan klasifikasi baru. Dengan menggunakan transfer learning, model dapat dioptimalkan untuk mengenali fitur-fitur yang relevan, seperti tekstur dan warna dari citra daun yang terinfeksi [5].

2.10. Penerapan pada Aplikasi Web

Dalam konteks klasifikasi citra penyakit tanaman, aplikasi berbasis web menawarkan solusi yang praktis dan mudah diakses oleh para petani. Penerapan model machine learning dalam sistem berbasis web memudahkan pengguna untuk mengunggah citra tanaman dan menerima hasil klasifikasi secara real-time. Penggunaan framework seperti Flask atau Django memungkinkan integrasi yang efisien antara model pembelajaran mesin dan antarmuka pengguna, sehingga memfasilitasi deteksi penyakit tanaman secara cepat dan efektif [7].

2.11. Pentingnya Aplikasi Web dalam Klasifikasi Citra

Aplikasi web menyediakan platform mudah diakses untuk klasifikasi penyakit tanaman. Pengguna, seperti petani, dapat mengunggah gambar tanaman dan mendapatkan hasil klasifikasi secara instan.

2.12. Framework untuk Pengembangan Aplikasi Web

Beberapa framework populer untuk mengembangkan aplikasi web berbasis AI adalah Flask, Django, dan Streamlit. Flask sering digunakan karena kesederhanaannya dalam integrasi dengan model machine learning seperti TensorFlow dan PyTorch.

2.13. Keunggulan Aplikasi Web untuk Pengguna

Aplikasi berbasis web memungkinkan akses yang lebih mudah dan cepat untuk mendiagnosis penyakit pada tanaman tanpa perlu instalasi perangkat lunak tambahan.

3. METODE PENELITIAN

Penelitian ini bertujuan untuk mendeteksi penyakit pada daun tebu menggunakan metode transfer learning. Perangkat keras yang digunakan adalah laptop Asus GL503GE dengan spesifikasi Intel Core i7-8750H, dan perangkat lunak meliputi Google Colaboratory, Python 3.x, serta framework PyTorch atau TensorFlow. Dataset yang digunakan diambil dari Kaggle.com dan terdiri dari gambar daun tebu sehat dan yang terinfeksi oleh berbagai penyakit seperti Red Rot, Red Stripe, Rust, dan Bacterial Blight, dengan total 500 gambar berukuran 128x128 piksel.

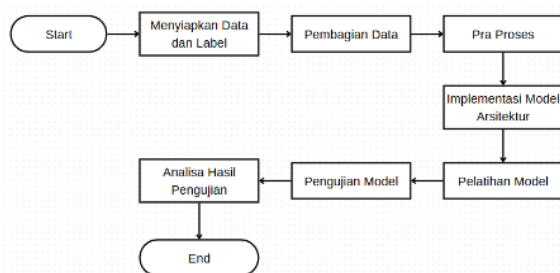
3.1. Pengumpulan Data

Data yang digunakan adalah gambar daun tebu, dibagi menjadi lima kategori: sehat, Red Rot, Red Stripe, Rust, dan Bacterial Blight. Setiap kategori terdiri dari 100 gambar.

3.2. Pra-Proses Data

Gambar diberi label dan diubah ukurannya menjadi 128x128 piksel agar dapat diproses lebih lanjut.

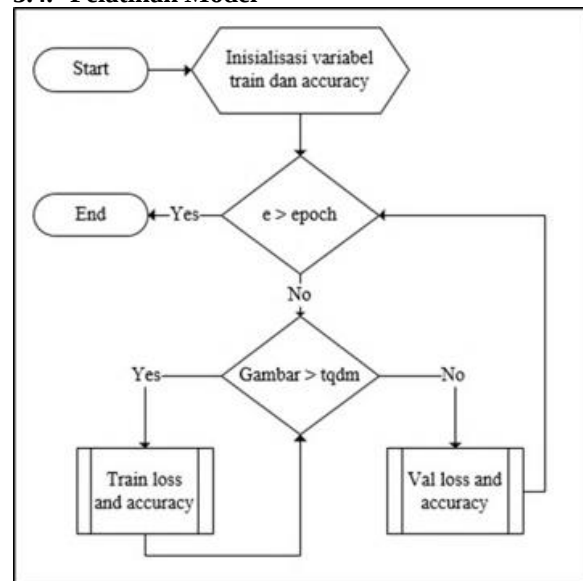
3.3. Perancangan Transfer Learning



Gambar 1. Flowchart perancangan metode

Tahap pertama pada perancangan metode transfer learning ini yaitu memuat dataset kemudian memuat library yang akan digunakan lalu dilanjutkan pada proses pembuatan model. Model Arsitektur yang digunakan pada penelitian ini yaitu AlexNet, DenseNet121, EfficientNet-B0, MobileNetV2, ResNet34 untuk mendapatkan nilai presentase akurasi di akhir penelitian. Tahap selanjutnya, memasukkan dataset ke dalam model transfer learning yang telah dibuat dan melakukan pelatihan model serta proses validasi model. Setelah itu, gunakan matplotlib untuk menampilkan grafik akurasi dan kehilangan, dan langkah terakhir adalah melakukan proses pengujian model dengan data acak dan menyelesaikannya.

3.4. Pelatihan Model

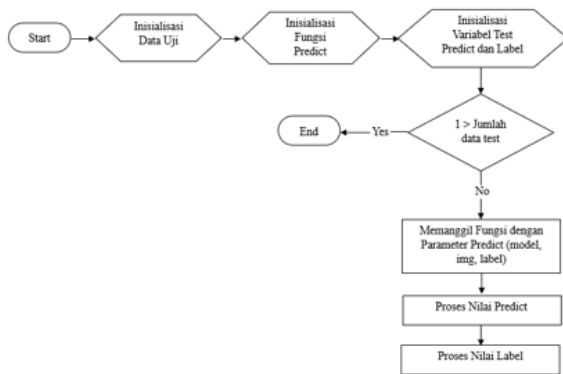


Gambar 2. Flowchart pelatihan model

Pada tahap pertama pelatihan model diawali dengan memasukkan inisialisasi variable train dan accuracy. Terdapat 2 variabel pada flowchart tersebut dimana variable train digunakan sebagai inisialisasi data latih dan variable accuracy digunakan untuk menginisialisasi data akurasi. Tqdm merupakan modul yang digunakan untuk python yang menampilkan table proses pengulangan yang sederhana. Pengulangan tersebut dilakukan hingga mendapatkan nilai e yang lebih besar dari epoch. Parameter yang dikustomisasi adalah epoch dan learning rate yang dimana parameter tersebut berguna dalam menentukan presentase akurasi pada penelitian ini.

3.5. Pengujian Model

Data uji digunakan untuk memprediksi jenis penyakit pada daun tebu. Model yang terbaik, berdasarkan akurasi dan kriteria lainnya, akan dipilih untuk implementasi aplikasi.



Gambar 3. Flowchart Pengujian Model

3.6. Skenario Uji Coba

Tahap ini dilakukan untuk mendapatkan hasil yang sesuai, berikut merupakan skenario uji coba yang dilakukan pada penelitian ini :

- Melakukan pengujian data latih, data validasi, dan data uji berupa citra penyakit RedRot, RedRust, dan daun yang sehat pada daun tebu.
- Memberikan parameter seperti pada tabel 1. dan variable uji transfer learning seperti pada tabel 2.
- Menggunakan 5 Arsitektur yaitu AlexNet, DenseNet121, EfficientNet-B0, MobileNetV2, ResNet34.
- Memperoleh hasil tingkat akurasi dari hasil data latih dan validasi lalu menampilkan hasil dari data yang telah di uji.

Tabel 1. Parameter

No	Parameter	Nilai Parameter
1	Jumlah kelas	5
2	Ukuran citra data latih	128 x 128
3	Jumlah Convolution Layer	36

Tabel 2. Variabel Uji Coba

Variabel	Nilai
Epoch	20
Batch Size	16
Learning Rate	1e-4 (0.0001) 1e-5 (0.00001)
Weight Decay	1e-4 (0.0001)
Data Latih dan Validasi	85%
Data Uji	15%
Optimizer	ADAM
Fungsi Aktivasi	Softmax

3.7. Implementasi Aplikasi Web

Model yang terlatih diintegrasikan ke dalam aplikasi web berbasis Python dengan framework Flask.

4. HASIL DAN PEMBAHASAN

4.1. Hasil Pengujian

Penelitian ini mengimplementasikan transfer learning untuk klasifikasi citra penyakit daun tebu dengan memanfaatkan lima arsitektur model deep learning: AlexNet, DenseNet121, EfficientNet-B0, MobileNetV2, dan ResNet34. Setiap arsitektur diuji

dengan dua nilai learning rate (0.0001 dan 0.00001) untuk melihat dampaknya terhadap performa model. Berdasarkan hasil pelatihan dan pengujian, berikut adalah ringkasan temuan utama:

4.2. Waktu Pelatihan dan Iterasi

- Arsitektur **EfficientNet-B0** menunjukkan waktu pelatihan tercepat dibandingkan dengan arsitektur lainnya. Keunggulan ini berasal dari pendekatan **compound scaling** yang mengoptimalkan kedalaman, lebar, dan resolusi gambar secara efisien. Hasilnya, EfficientNet-B0 dapat mencapai konvergensi lebih cepat dengan waktu iterasi yang lebih singkat. **MobileNetV2**, meskipun sedikit lebih lambat, juga menunjukkan waktu pelatihan yang kompetitif. Kedua model ini sangat efisien, terutama ketika diterapkan pada perangkat dengan keterbatasan sumber daya.
- Model **DenseNet121** memiliki waktu pelatihan yang paling lama. Arsitektur ini menggunakan **dense blocks** yang menghubungkan semua lapisan konvolusi sebelumnya, yang membuatnya lebih dalam dan kompleks. Meskipun membutuhkan waktu lebih lama untuk pelatihan, DenseNet121 memberikan hasil yang sangat baik dalam hal akurasi, terutama saat menggunakan learning rate yang lebih tinggi (0.0001).

4.3. Akurasi dan Loss

- DenseNet121** dengan learning rate **0.0001** mencapai akurasi pelatihan hampir 100%. Hal ini menunjukkan bahwa model ini sangat efektif dalam mengklasifikasikan gambar dengan data yang lebih banyak dan lebih kompleks. Model ini dapat belajar dengan sangat baik dari dataset dan dapat menggeneralisasi dengan sangat baik pada data yang belum pernah dilihat sebelumnya. Ini menjadikannya pilihan yang sangat baik ketika akurasi sangat diutamakan.
- EfficientNet-B0** menunjukkan waktu pelatihan yang cepat, tetapi dengan **loss yang lebih tinggi** dibandingkan model lainnya. Meskipun model ini cepat, nilai **loss** yang lebih tinggi menandakan bahwa model cenderung sedikit lebih kesulitan dalam mengoptimalkan bobot pada data pelatihan. Loss yang tinggi mengindikasikan bahwa meskipun model dapat belajar lebih cepat, konvergensinya tidak sebaik model lain seperti DenseNet121.
- Pengaruh Learning Rate 0.00001**: Penggunaan learning rate yang lebih kecil (0.00001) menunjukkan **penurunan akurasi** dan **peningkatan loss** pada sebagian besar model. Misalnya, pada **EfficientNet-B0**, model ini tidak dapat mempertahankan performa yang baik dengan learning rate lebih kecil. Hal ini disebabkan oleh proses pembelajaran yang lebih lambat, di mana model membutuhkan lebih

banyak iterasi untuk mencapai konvergensi. Akibatnya, **loss** lebih tinggi dan **akurasi** lebih rendah dibandingkan saat menggunakan learning rate yang lebih besar (0.0001).

4.4. Confusion Matrix

- Hasil confusion matrix untuk **AlexNet** dan **ResNet34** dengan learning rate 0.0001 menunjukkan klasifikasi yang sangat baik, dengan lebih dari 99% gambar diklasifikasikan dengan benar.
- MobileNetV2** dan **DenseNet121** memiliki sedikit kesalahan klasifikasi, namun secara keseluruhan tetap memberikan hasil yang baik.
- Penggunaan learning rate 0.00001 mengurangi performa model, terutama pada **EfficientNet-B0**, yang menghasilkan banyak kesalahan klasifikasi.

Tabel 3. Hasil Performa Pelatihan
(Learning Rate 0.0001)

Parameter	Alex Net	Dense Net1 21	Efficient Net-B0	Mobile Net V2	ResNet34
Waktu/Iterasi	4.79s	7.13s	4.18s	4.94s	6.62s
Train Loss	0.260	0.329	0.595	0.279	0.262
Train Accuracy	0.893	0.875	0.812	0.907	0.895
Validation Loss	0.521	0.430	0.583	0.525	0.309
Validation Accuracy	0.867	0.876	0.800	0.891	0.874

Berdasarkan hasil pada tabel tersebut dengan menggunakan nilai learning rate sebesar 0.001, waktu yang dibutuhkan model untuk 1 kali iterasi bervariasi antar arsitektur, dapat disimpulkan bahwa EfficientNet-B0 memiliki waktu iterasi paling cepat, namun memiliki train loss dan validation loss yang relatif lebih tinggi. MobileNetV2 dan ResNet34 menunjukkan keseimbangan yang baik antara akurasi dan loss, dengan MobileNetV2 mencapai train accuracy tertinggi. Sedangkan DenseNet121 memiliki waktu iterasi terlama namun menunjukkan performa validasi yang baik dengan validation loss terendah setelah ResNet34.

Tabel 4. Hasil Performa Pelatihan
(Learning Rate 0.00001)

Parameter	Alex Net	Dense Net1 21	Efficient Net-B0	Mobile Net V2	ResNet34
Waktu/Iterasi	3.72s	4.62s	4.13s	4.35s	5.03s
Train Loss	0.599	0.959	1.284	0.685	0.711
Train Accuracy	0.810	0.728	0.584	0.786	0.772
Validation Loss	0.499	0.909	1.299	0.846	0.818
Validation Accuracy	0.815	0.670	0.584	0.677	0.720

Berdasarkan hasil pada tabel diatas dengan menggunakan nilai learning rate sebesar 0.00001, waktu yang dibutuhkan model untuk 1 kali iterasi

bervariasi antar arsitektur, dapat disimpulkan bahwa AlexNet memiliki waktu iterasi tercepat dengan performa train loss dan validation loss yang lebih baik dibandingkan model lain pada learning rate ini. EfficientNet-B0 memiliki waktu iterasi yang cepat, namun menunjukkan nilai loss tertinggi dan akurasi terendah. MobileNetV2 dan ResNet34 menunjukkan performa yang seimbang antara akurasi dan loss, sedangkan DenseNet121 memiliki waktu iterasi lebih lama dengan performa validasi yang kurang optimal.

Tabel 5. Hasil Performance Metrics Dengan Menggunakan Nilai Learning Rate 0.0001 Pada Setiap Arsitektur

Arsitektur	Akurasi (%)	Precision (%)	Recall (%)	Waktu
AlexNet	99%	99%	99%	2204.15s
DenseNet121	100%	100%	100%	3280.47s
EfficientNet-B0	99%	99%	99%	1927.31s
MobileNetV2	96%	96%	96%	2273.44s
ResNet34	99%	99%	99%	3047.33s

Berdasarkan hasil pengujian dari berbagai arsitektur model pada tabel diatas, didapatkan analisis, secara keseluruhan, DenseNet121 memberikan hasil terbaik dalam hal akurasi dan ketelitian klasifikasi, namun dengan trade-off waktu pelatihan yang lebih lama. Sedangkan EfficientNet-B0 memiliki waktu pelatihan paling cepat dengan performa yang hampir setara dengan AlexNet dan ResNet34. MobileNetV2 cocok digunakan jika kebutuhan waktu pelatihan lebih diperhatikan dengan sedikit pengorbanan pada performa.

Tabel 6. Hasil Performance Metrics Dengan Menggunakan Nilai Learning Rate 0.00001 Pada Setiap Arsitektur

Arsitektur	Akurasi (%)	Precision (%)	Recall (%)	Waktu
AlexNet	92%	92%	92%	2137.68s
DenseNet121	85%	90%	85%	2656.23s
EfficientNet-B0	75%	76%	75%	2287.86s
MobileNetV2	89%	90%	89%	2387.35s
ResNet34	91%	91%	91%	2842.56s

Berdasarkan hasil pengujian dari berbagai arsitektur model pada tabel diatas, didapatkan analisis, bahwa AlexNet dan ResNet34 memberikan keseimbangan terbaik antara akurasi dan waktu pelatihan, sementara EfficientNet-B0 memiliki performa paling rendah meskipun waktu pelatihannya relatif cepat. DenseNet121 dan MobileNetV2 menawarkan alternatif dengan precision yang cukup baik, namun dengan akurasi sedikit lebih rendah dibandingkan AlexNet dan ResNet34. Hasil F1 – Score dengan menggunakan nilai learning rate 0.0001 pada tiap Arsitektur yang telah di uji ditampilkan pada tabel 4.10 sebagai berikut :

Tabel 7. Hasil F1 - Score Dengan Menggunakan Nilai Learning Rate 0.0001

Kelas	AlexNet	Dense Net121	Efficient Net-B0	Mobile NetV2	ResNet34
Rust	100%	100%	100%	100%	100%
Red Stripe	100%	100%	100%	100%	100%
Red Rot	97%	100%	97%	90%	100%
Healthy	100%	100%	100%	97%	97%
Bacterial Blight	97%	100%	97%	93%	97%

Hasil F1 – Score dengan menggunakan nilai learning rate 0.00001 pada tiap Arsitektur yang telah di uji ditampilkan pada tabel 4.11 sebagai berikut :

Tabel 8. Hasil F1 - Score Dengan Menggunakan Nilai Learning Rate 0.00001

Kelas	AlexNet	Dense Net121	Efficient Net-B0	Mobile NetV2	ResNet34
Rust	100%	89%	55%	100%	93%
Red Stripe	100%	91%	88%	100%	97%
Red Rot	84%	80%	67%	82%	93%
Healthy	90%	89%	77%	83%	90%
Bacterial Blight	87%	79%	79%	81%	80%

Learning rate 0.0001 menghasilkan performa F1-Score terbaik dan paling konsisten di seluruh arsitektur dan kelas, dengan skor sangat tinggi mendekati sempurna. Penurunan learning rate menjadi 0.00001 menyebabkan penurunan signifikan dalam performa, terutama pada EfficientNet-B0, yang menunjukkan bahwa pemilihan learning rate sangat berpengaruh terhadap kualitas klasifikasi model. Arsitektur seperti AlexNet dan MobileNetV2 menunjukkan ketahanan yang lebih baik terhadap perubahan learning rate dibandingkan yang lain.

F1-Score

- Pada **learning rate 0.0001**, **DenseNet121** mencapai nilai F1-score hampir sempurna (100%) untuk semua kelas.
- **AlexNet** dan **ResNet34** menunjukkan performa yang sangat baik dengan nilai F1-score yang tinggi, terutama pada kelas penyakit seperti *Rust*, *Red Stripe*, dan *Healthy*.

4.5. Implementasi Web Aplikasi

Setelah pelatihan model, aplikasi berbasis web dikembangkan untuk memungkinkan petani atau praktisi pertanian mengunggah gambar daun tebu untuk melakukan prediksi penyakit secara real-time. Aplikasi ini menggunakan framework Flask, yang memungkinkan integrasi mudah dengan model deep learning dan memberikan antarmuka yang mudah digunakan untuk pengguna. Aplikasi web ini bertujuan untuk memberikan kemudahan akses bagi pengguna, khususnya petani atau praktisi pertanian, dalam

melakukan prediksi penyakit pada daun tebu secara cepat dan interaktif melalui perangkat komputer atau smartphone tanpa perlu instalasi perangkat lunak khusus.

4.6. Pemilihan Framework dan Teknologi

Aplikasi web dikembangkan menggunakan framework Flask, yaitu framework Python yang ringan dan mudah digunakan untuk membangun API dan aplikasi web sederhana. Flask memungkinkan integrasi yang efisien dengan model deep learning yang sudah dilatih menggunakan PyTorch. Selain itu, Flask mendukung pengelolaan request secara real-time sehingga prediksi dapat dilakukan langsung setelah pengguna mengunggah gambar.

4.7. Arsitektur Sistem Web

Sistem aplikasi web ini terdiri dari beberapa komponen utama, yaitu:

- Frontend: Halaman web sederhana yang menyediakan fitur upload gambar daun tebu oleh pengguna dan menampilkan hasil prediksi penyakit.
- Backend: Server Flask yang menerima gambar, melakukan preprocessing yang sama seperti pada saat pelatihan (resize, normalisasi), kemudian memanggil model deep learning untuk menghasilkan prediksi kelas penyakit.
- Model Deep Learning: Model transfer learning yang telah dilatih dan disimpan dalam format .pth dimuat di server backend dan digunakan untuk inferensi.

4.8. Proses Prediksi

Proses prediksi pada aplikasi web berjalan sebagai berikut:

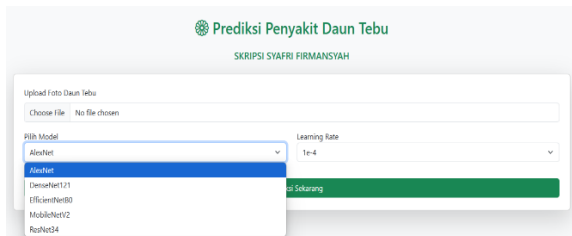
- Pengguna mengunggah gambar daun tebu melalui interface web.
- Gambar yang diterima oleh server diubah ukurannya sesuai dengan input model (misalnya 128 x 128 piksel) dan dinormalisasi dengan mean dan std yang sama seperti saat pelatihan.
- Gambar diproses oleh model deep learning untuk mengklasifikasikan penyakit daun tebu ke salah satu dari lima kelas yang sudah ditentukan.
- Hasil prediksi berupa label penyakit dan tingkat keyakinan (confidence) dikirim kembali dan ditampilkan ke pengguna.

4.9. Pengujian dan Evaluasi Aplikasi Web

Aplikasi diuji untuk memastikan bahwa proses upload, preprocessing, prediksi, dan tampilan hasil berjalan dengan baik dan cepat. Hasil pengujian menunjukkan bahwa aplikasi mampu melakukan prediksi dengan latency rata-rata kurang dari beberapa detik, yang sudah cukup efisien untuk penggunaan sehari-hari. Untuk hasil dari web aplikasi yang telah dibuat ditampilkan pada gambar di bawah ini :



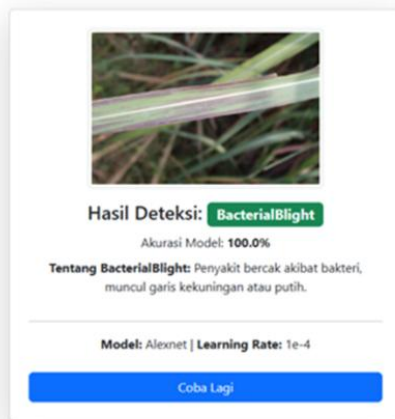
Gambar 4. Tampilan halaman utama aplikasi web



Gambar 5. Menu Pemilihan Arsitektur



Gambar 6. Menu Pemilihan Learning rate



Gambar 7. Hasil prediksi penyakit daun tebu yang ditampilkan di aplikasi web

4.10. Evaluasi Aplikasi Web

Waktu respons aplikasi rata-rata kurang dari beberapa detik, membuatnya praktis untuk digunakan di lapangan. Aplikasi ini memungkinkan prediksi interaktif yang langsung memberikan hasil kepada pengguna, mempermudah diagnosis penyakit daun tebu. Hasil pengujian menunjukkan bahwa aplikasi mampu melakukan prediksi dengan latency rata-rata kurang dari beberapa detik, yang sudah cukup efisien untuk penggunaan sehari-hari.

5. KESIMPULAN DAN SARAN

Berdasarkan hasil penelitian yang telah dilakukan dapat disimpulkan bahwa, penelitian ini berhasil mengimplementasikan metode transfer learning menggunakan lima arsitektur CNN populer, yaitu AlexNet, DenseNet121, EfficientNet-B0, MobileNetV2, dan ResNet34, untuk klasifikasi citra penyakit daun tebu dengan 5 kelas berbeda. Dengan nilai learning rate 0.0001 memberikan performa pelatihan yang lebih baik dibandingkan nilai learning rate 0.00001, dengan hasil akurasi, precision, recall, dan F1-score yang lebih tinggi dan proses pelatihan yang lebih stabil serta lebih cepat konvergen. Dari kelima arsitektur yang diuji, DenseNet121 menunjukkan hasil akurasi terbaik hingga 100% pada data validasi dengan trade-off waktu pelatihan paling lama. Implementasi model terbaik dalam aplikasi web berbasis Flask berhasil menyediakan sistem klasifikasi penyakit daun tebu yang mudah diakses dan memberikan prediksi dengan latency yang rendah. Meskipun dataset yang digunakan relatif kecil (500 gambar), hasil yang diperoleh menunjukkan bahwa transfer learning merupakan pendekatan yang efektif untuk klasifikasi citra penyakit tanaman dengan performa yang baik.

Penulis memberikan beberapa saran untuk penelitian selanjutnya dan pengembangan aplikasi lebih lanjut, yakni, untuk meningkatkan kemampuan generalisasi model dan mengurangi risiko overfitting, disarankan untuk menggunakan dataset yang lebih besar dan bervariasi. Pengembangan aplikasi web dapat ditingkatkan dengan menambahkan fitur real-time detection, feedback pengguna, dan antarmuka yang lebih interaktif agar lebih mudah digunakan oleh petani dan praktisi. Perlu dilakukan pengujian robustness model terhadap variasi kondisi gambar seperti pencahayaan berbeda, noise, atau sudut pengambilan gambar untuk meningkatkan keandalan sistem dalam kondisi nyata.

DAFTAR PUSTAKA

- [1] Y. V. Sari, Z. Muallifah, and A. Fanani, "Klasifikasi Kualitas Air Menggunakan Metode Extreme Learning Machine (ELM)," *J. JUPITER*, vol. 15, no. 2, pp. 983–994, 2023, [Online]. Available: <https://jurnal.polsri.ac.id/index.php/jupiter/article/view/6995>
- [2] G. H. A. Panjaitan and F. Simatupang, "Pemodelan Klasifikasi Penyakit Daun Tanaman Tomat dengan Convolutional Neural Network Algorithm," *KLIK Kaji. Ilm. Inform. dan Komput.*, vol. 4, no. 5, pp. 2667–2675, 2024, doi: 10.30865/klik.v4i5.1646.
- [3] N. A. E. Putra and S. Agustin, "Klasifikasi Kematangan Tebu Berdasarkan Tekstur Batang Menggunakan Metode Naïve Bayes," *Indexia*, vol. 2, no. 2, pp. 23–28, 2021, doi: 10.30587/indexia.v2i2.2559.

- [4] T. Yulianti, "STATUS DAN STRATEGI TEKNOLOGI PENGENDALIAN PENYAKIT UTAMA TEBU DI INDONESIA Status and Control Strategy of Important Sugarcane Diseases In ...," *Perspektif*, vol. 19, no. 1, pp. 1–16, 2020, [Online]. Available: <http://ejurnal.litbang.pertanian.go.id/index.php/sp/article/view/11057>
- [5] N. Awalia, "Identifikasi Penyakit Leaf Mold Pada Daun Tomat Menggunakan Model Densenet121 Berbasis Transfer Learning," *J. Ilm. Ilmu Komput.*, vol. 8, no. 1, pp. 49–52, 2022, doi: 10.35329/jiik.v8i1.212.
- [6] F. Rochman and H. Junaedi, "Klasifikasi objek saat ini merupakan salah satu studi 24," *J. Syntax Admiration*, vol. 1, no. 6, pp. 672–679, 2020.
- [7] S. T. Damanik, A. Setiawan, T. Simanjuntak, A. A. Tumanggor, and F. Ramadhani, "Analisis Deteksi Tepi untuk Mengidentifikasi Jenis Mobil," *Explor. J. Sist. Inf. dan Telemat.*, vol. 14, no. 1, p. 103, 2023, doi: 10.36448/jsit.v14i1.3082.
- [8] E. Retnoningsih and R. Pramudita, "Mengenal Machine Learning Dengan Teknik Supervised Dan Unsupervised Learning Menggunakan Python," *Bina Insa. Ict J.*, vol. 7, no. 2, p. 156, 2020, doi: 10.51211/biict.v7i2.1422.
- [9] F. Mashuri, "Implementasi Transfer Learning Dalam Mendeteksi Penyakit Pada Daun Gandum," *Nuansa Inform.*, vol. 16, no. 1, pp. 66–77, 2022, doi: 10.25134/nuansa.v16i1.4702.
- [10] M. Ramadhan, D. I. Mulyana, and M. B. Yel, "Optimasi Algoritma Cnn Menggunakan Metode Transfer Learning Untuk Klasifikasi Citra X-Ray Paru-Paru Pneumonia Dan Non-Pneumonia," *J. Tek. Inform. Kaputama*, vol. 6, no. 2, pp. 670–679, 2022.