

Detection of Porang Plant Diseases and Pests (*Amorphophallus Muelleri*) Based on Leaf Imagery Utilizing DCNN Transfer Learning

Deteksi Penyakit dan Hama Tanaman Porang (*Amorphophallus Muelleri*) Berdasarkan Citra Daun Memanfaatkan DCNN *Transfer Learning*

Miftahuz Zuhan¹, Yosi Kristian²

^{1,2}Magister Teknologi Informasi, Institut Sains Dan Teknologi Terpadu Surabaya

E-mail: *¹daehanaqila@gmail.com, ²yosi@stts.edu

Abstract – Processed products from the corm of the porang plant (*Amorphophallus muelleri*) are always in demand in the Asian region, including Japan, China, Korea, and Australia. The corms can be used as raw materials in the cosmetics industry and have the potential to prevent various human diseases due to their high glucomannan content. However, many farmers face various diseases, such as leaf rot disease, mosaic virus (*konjac mosaic*), and pest attacks on porang plants, in order to obtain high-quality porang corms. In this study, a deep learning architecture is proposed for the classification of leaf diseases in porang plants. The performance of the Custom CNN model is compared with other deep learning models. All models are trained on the original dataset and augmented data of 1000 images. Transfer Learning approach is used to train all deep learning models. The dataset testing results show that the EfficientNetV2M architecture model achieves the highest score compared to other deep learning models on the augmented dataset with an accuracy of 98.44%. ResNet50 on the original dataset with a value of 97.66% on the original dataset, while Custom CNN with an accuracy of 89.06% and 85.94% for all datasets.

Keywords — *Amorphophallus* diseases, deep convolutional neural network, deep transfer learning

Abstrak –Produk olahan dari umbi tanaman porang (*Amorphophallus muelleri*) selalu diminati di kawasan Asia: Jepang, China, Korea, serta negara kawasan Australia. Umbi tanaman porang dapat digunakan sebagai bahan baku industri kosmetik dan juga memiliki potensi untuk mencegah berbagai penyakit manusia, karena memiliki kandungan glukoman yang tinggi. Untuk mendapatkan umbi porang yang berkualitas baik, banyak petani menghadapi berbagai penyakit seperti penyakit busuk daun, virus mozaik (*mosaik konjac*) dan serangan hama pada daun tanaman porang. Dalam studi ini, diajukan sebuah arsitektur *deep learning* untuk klasifikasi penyakit daun pada tanaman porang. Kinerja dari model CNN *Custom* dibandingkan dengan model *deep learning* lainnya. Semua model dilatih pada kumpulan data asli dan data augmentasi dari 1000 gambar. Pendekatan *Transfer Learning* digunakan untuk melatih semua model *deep learning*. Hasil pengujian *dataset* menunjukkan bahwa model arsitektur *EfficientNetV2M* mencapai skor tertinggi dibandingkan dengan model *deep learning* lainnya pada *dataset* augmentasi dengan akurasi sebesar 98,44%. Sedang *ResNet50* pada *dataset* asli dengan nilai 97,66% pada *dataset* asli, sedangkan CNN *Custom* dengan akurasi sebesar 89,06% dan 85,94% untuk semua *dataset*.

Kata Kunci — penyakit tanaman porang, *deep convolutional neural network*, *deep transfer learning*

1. PENDAHULUAN

Amorphophallus muelleri Blume atau yang lebih dikenal dengan nama Porang, adalah tumbuhan asli Indonesia yang saat ini populer di kawasan Asia; Jepang, Cina, Korea, dan di benua Australia. Umbi Porang merupakan komoditas ekspor yang memiliki harga yang sangat menjanjikan bagi para petani Porang di Indonesia [1]. Awalnya, para petani hanya menjual Porang dalam bentuk umbi mentah atau irisan Porang kering, sehingga nilai ekonomisnya masih kurang memuaskan. Namun, setelah diketahui banyak manfaat yang terkandung pada umbi Porang, seperti untuk mencegah dan mengurangi berbagai jenis penyakit manusia, menjadi bahan baku industri kecantikan, dan bahan pengganti beras, produksi olahan Porang di Madiun dan Mojokerto meningkat dengan adanya pabrik pengolahan Porang yang didirikan oleh Kementerian Pertanian Indonesia.

Namun, untuk meningkatkan produktivitas umbi Porang, tidak dapat dipungkiri bahwa tanaman Porang juga mengalami berbagai penyakit dan gangguan hama [2]. Beberapa penyakit yang sering terjadi pada tanaman Porang adalah busuk kaki (*root fot*) yang disebabkan oleh *Rhizotonia Solani*, hawar daun oleh virus *Phytophthora Colocasiae*, pembusukan umbi (*root rot*) oleh jamur *Phytium Helicoides* dan *Slerotium Rolfsii*, pembusukan batang (*soft rot*) oleh virus *Erwinia Carotovora* serta daun belang putih oleh *Konjac Mosaic Virus*, dan gangguan hama seperti ulat grayak dan *nematoda*. Jika tidak ditangani dengan baik, penyakit dan hama ini dapat menurunkan nilai produksi umbi Porang.

Namun, bidang teknologi juga turut berperan dalam meningkatkan produktivitas umbi Porang, salah satunya adalah teknologi informatika dengan memanfaatkan citra digital (*image processing*) untuk mengidentifikasi penyakit dan hama pada tanaman Porang [3]. Salah satu aplikasi penelitian dengan citra digital (*image processing*) adalah untuk mendeteksi dan mengidentifikasi penyakit dan hama pada tanaman Porang serta meningkatkan hasil produksi di bidang pertanian dan perkebunan [3]. Telah dilakukan beberapa penelitian dalam mendeteksi dan mengklasifikasikan hama dan penyakit pada tanaman menggunakan metode *image processing*, *machine learning*, dan *deep learning*. Hasil dari proses deteksi tersebut menunjukkan presisi dan akurasi yang baik. Salah satu penelitian yang berhasil dalam mendeteksi dan mendiagnosis penyakit pada tanaman dengan menggunakan *deep convolutional neural networks* (DCNN) untuk mendiagnosis penyakit pada tanaman padi dengan *dataset* yang terdiri dari 500 gambar daun padi yang terserang penyakit daun, yang kemudian dibagi menjadi 10 kelas [4]. Setelah diolah dan dianalisis dengan *digital image processing*, citra tersebut dikompres menjadi ukuran 512x512 piksel. Pada proses pengujian, *Convolutional Neural Network* (CNN) dengan *base model LeNet* dan *Alexnet* digunakan dengan *filter size* yang berbeda untuk memperoleh performa terbaik, dan berhasil memperoleh nilai akurasi sebesar 94,9%. Pada Penelitian menggunakan CNN untuk pendeteksian hama dan penyakit pada tanaman tomat dengan menggunakan tiga model *deep learning* yaitu *Faster R-CNN*, *SSD*, dan *ResNet* [5]. Pada *Faster R-CNN* dilakukan dengan 2 tahap yaitu *feature extraction* dan *intermediate*, sedangkan *SSD* menggunakan *feed-forward convolutional network* yang dihasilkan oleh *bounding box* dengan ukuran dan skor tetap untuk kelas objek di setiap *box*. Penggunaan *ResNet-50* dan *ResNet-101* sebagai fitur ekstraktor dihasilkan akurasi 75.37% hingga 85.98%.

Penelitian lain serupa telah dilakukan untuk mendeteksi dan mengklasifikasikan penyakit pada tanaman jagung dengan menggunakan CNN dan telah dimodifikasi dan dilatih menggunakan data dari *PlantVillage*. Model CNN yang dibuat dapat mengenali tiga jenis penyakit pada daun jagung, yaitu hawar daun jagung utara (*Exserohilum*), karat biasa (*Puccinia sorghi*), dan bercak daun abu-abu (*Cercospora*). Model ini dibuat dengan menggunakan 600 *input*, Model CNN yang digunakan memiliki 50 *hidden layer* dan setiap kategori terdiri dari 100 gambar, dengan rasio pembagian *dataset* dengan proporsi 70% untuk data latih dan 30% untuk data uji. Dari hasil pengujian menunjukkan bahwa CNN dapat mengenali dan mengklasifikasikan penyakit daun jagung dengan akurasi keseluruhan sebesar 92,85%. Pada pengujian untuk setiap jenis penyakit terpisah, akurasinya adalah 99,9% untuk hawar daun, 91% untuk bercak daun abu-abu, 87% untuk karat biasa, dan 93,5% untuk daun jagung yang sehat [6].

Penelitian ini untuk mengusulkan sistem yang memudahkan petani Porang dalam mengidentifikasi dan mengkategorikan daun ke dalam lima jenis, yaitu daun sehat atau normal, penyakit busuk daun, bercak daun, daun bergaris (*mosaik konjac*), dan serangan hama ulat grayak dan belalang.

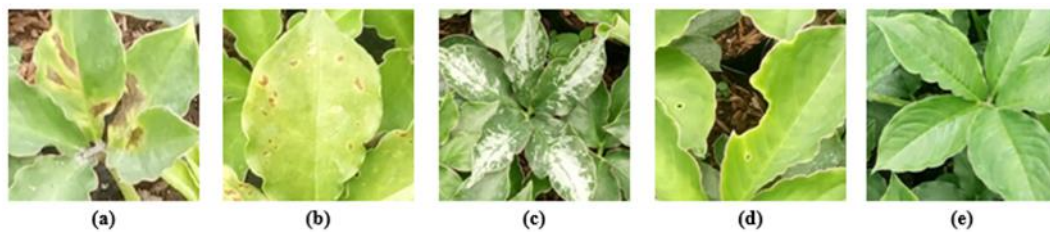
Pengenalan dilakukan melalui penggunaan algoritma *Deep Convolution Neural Network Transfer Learning*, teknik *deep learning* yang cocok untuk dipakai dalam pengolahan citra digital [7]. Kontribusi utama dari artikel ini dirangkum sebagai berikut: (1) *Dataset* citra daun porang berpenyakit menggunakan data primary dengan jumlah terbatas. Untuk memberikan data citra yang cukup bagi model DCNN, dilakukan dengan augmentasi gambar untuk meningkatkan kemampuan generalisasi model DCNN dan menghindari masalah overfitting. (2) Model DCNN yang dibangun, diusulkan untuk pendeteksian dan klasifikasi *end-to-end* untuk penyakit daun pada tanaman Porang. Dengan penambahan dan modifikasi pada *layer* konvolusi (*dense*), *dropout* serta *learning rate* untuk meningkatkan akurasi model DCNN pada penyakit dan hama pada tanaman porang. Diharapkan penelitian ini dapat menjadi referensi bagi para peneliti dan praktisi serta memicu pengembangan penelitian lebih lanjut di masa depan.

2. METODE PENELITIAN

2.1. Dataset

Penelitian ini menggunakan *dataset* berupa citra daun tanaman porang yang terkena hama dan penyakit. Data diambil langsung dari kebun dan ladang petani porang. Terdapat lima jenis daun tanaman porang yang digunakan sebagai sampel, yaitu busuk daun (*late blight*), bercak coklat (*early blight*), bercak putih (*mosaik konjac*), serangan serangga (*insects pest*), dan daun sehat (*healthy*) [8].

Sebanyak 1000 citra daun porang dikumpulkan dan diklasifikasikan ke dalam lima kategori dengan setiap kelas memiliki setidaknya 50-200 gambar untuk pelatihan. Data tersebut berisi gambar dengan berbagai ukuran dan resolusi, dan harus memiliki label untuk setiap kelas karena metode yang digunakan adalah *supervised learning*. Selain itu, metode augmentasi digunakan untuk memperbanyak jumlah data dengan berbagai variasi, tanpa mengubah kategori dataset yang ada [9].



Gambar 1. Contoh penyakit dan hama pada daun porang: a) Hawar daun (*late blight*), b) Karat/bercak daun (*early blight*), c) Mosaic (*konjac mosaic*), d) Terserang hama (*insects*) serta e) Daun sehat (*healthy*)

2.2. Model CNN

Performa arsitektur CNN yang digunakan dalam penelitian ini dengan membandingkan arsitektur CNN normal dengan arsitektur *transfer learning* dengan *pre-trained* seperti VGG16, ResNet50, dan EfficientNetV2M.

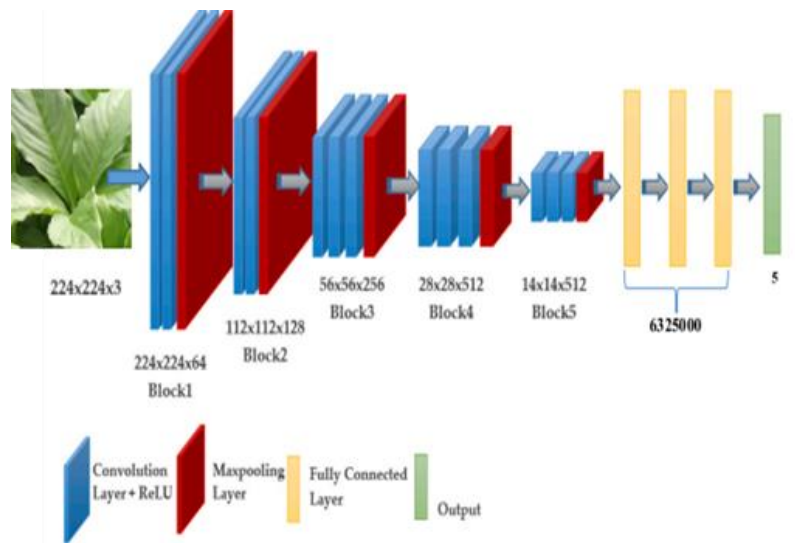
2.3. VGG16

VGG16, yang memenangkan ILSVRC-2014, merupakan sebuah arsitektur CNN dengan sekitar 138 juta parameter [10]. Salah satu fitur yang paling mencolok dari arsitektur ini adalah penggunaan jumlah *hyperparameter* yang besar. Selain itu, arsitektur ini memiliki lapisan konvolusi yang sama dengan filter 3x3 dengan *strade* 1, serta *padding* dan *max pooling layers* yang sama, menggunakan filter 2x2 dan *strade* yang sama dengan 2. Gambaran arsitektur VGG16 memiliki jumlah konvolusi dan urutan lapisan penyusunan yang maksimum serta konsisten di semua arsitektur (lihat Gambar 2) [11]. Arsitektur ini terdiri dari 16 lapisan dan lapisan *input* digunakan untuk menangkap

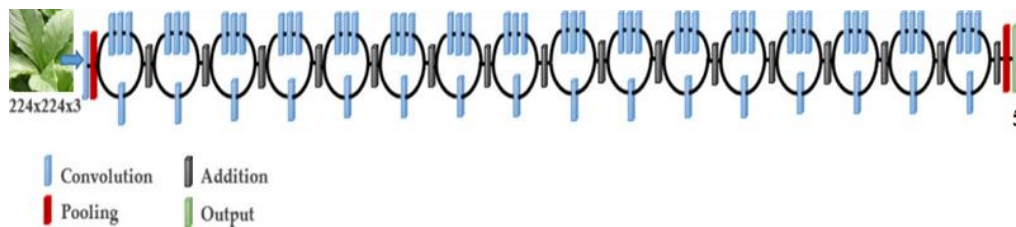
gambar dengan ukuran 224x224 piksel. Terdapat 3 lapisan *Fully Connected*, dua di antaranya menggunakan ReLU dan yang terakhir menggunakan aktivasi *Softmax*.

2.3.1. *ResNet50*

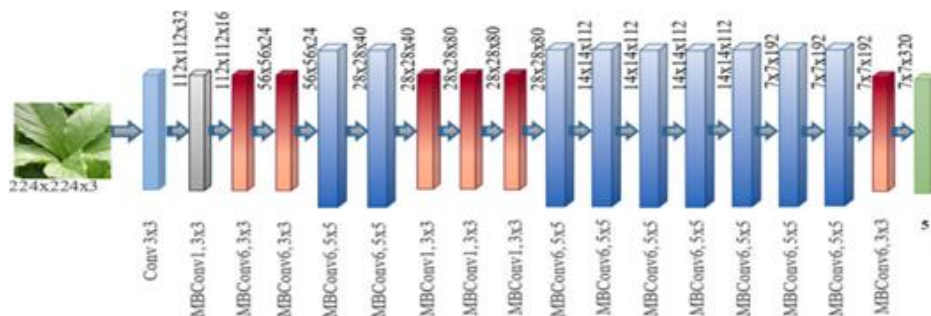
Arsitektur *ResNet50*, yang memenangkan kompetisi ILSVRC pada tahun 2015, merupakan sebuah model CNN yang diusulkan untuk mengatasi permasalahan dengan banyak lapisan non-linear. [12]. Model ini terdiri dari 50 lapisan yang di-*stack* satu sama lain untuk membentuk sebuah jaringan saraf akhir yang terdiri dari 48 layer konvolusi, 1 layer *average pooling*, dan 1 layer *maximum pooling*. Seperti halnya yang disampaikan oleh Gambar 2, arsitektur VGG16, *ResNet50* dan *EfficientNet* juga menggunakan filter 3x3 dengan ukuran gambar 224x224 piksel.



a



b



c

Gambar 2. Contoh penyakit dan hama pada daun porang: a) Hawar daun (*late blight*), b) Karat/bercak daun (*early blight*), c) *Mosaic* (*konjac mosaic*), d) Terserang hama (*insects*) serta e) Daun sehat (*healthy*)

2.3.2. EfficientNetV2M

Keberhasilan *AlexNet* dalam memenangkan kompetisi *ImageNet* 2012 menunjukkan bahwa model yang dibuat dengan *dataset ImageNet* menjadi lebih besar dan kompleks, tetapi kurang efisien dalam hal beban komputer. Model *EfficientNetV2M* merupakan salah satu model terancang yang mampu mencapai akurasi hingga 84,4% pada masalah klasifikasi *ImageNet* dengan parameter 66 M. Model *EfficientNetV2M* memiliki delapan model antara B0 hingga B7, semakin bertambahnya nomor model, jumlah parameter yang dihitung tidak meningkat secara signifikan, namun akurasi yang dihasilkan secara signifikan pesat. Model *EfficientNetV2M* menggunakan ReLU (*Rectifier Linear Unit*) sebagai fungsi aktivasi. [13].

Permasalahan utama dalam merancang CNN untuk semua jaringan saraf adalah penskalaan model, yaitu dengan cara meningkatkan ukuran model agar memberikan akurasi yang lebih tinggi dan lebih baik. *EfficientNetV2M* adalah arsitektur jaringan saraf canggih dan metode penskalaan yang terdiri dari tiga komponen: kedalaman (*depth*), lebar (*width*), dan resolusi (*resolution*). Penambahan ketiga komponen tersebut dilakukan dengan sangat teratur sehingga jumlah parameter menjadi lebih sedikit dan waktu proses menjadi lebih cepat, sementara akurasi masih terjaga dengan baik. [13].

EfficientNetV2M menggunakan lapisan *MBCConv* dan teknik penskalaan sederhana namun efektif dengan faktor ϕ untuk mengontrol sumber daya yang tersedia dan alokasinya ke kedalaman, lebar, dan resolusi jaringan. [14]. Penambahan *depth*, *width*, atau *resolution* akan meningkatkan anggaran FLOPS. Arsitektur ini memiliki *deep convolution* yang dapat dipisahkan untuk mengurangi beban komputasi. Teknik penskalaan yang sederhana namun sangat efektif dengan menggunakan beberapa faktor majemuk (ϕ) pada penskalaan lebar, kedalaman, dan resolusi jaringan secara seragam dengan menggunakan pedoman Persamaan 1 hingga Persamaan 5.

$$\text{depth: } d = \alpha ^ \phi \dots\dots\dots(1)$$

$$\text{width: } w = \beta ^ \phi \dots\dots\dots(2)$$

$$\text{resolution: } r = \gamma ^ \phi \dots\dots\dots(3)$$

$$\text{s.t. } \alpha .\beta ^2.\gamma ^2 \approx 2 \dots\dots\dots(4)$$

$$\alpha \geq 1, \beta \geq 1, \gamma \geq 1 \dots\dots\dots(5)$$

Keterangan :

d = *dept*

w = *width*

R = *resolution*.

ϕ adalah faktor yang ditentukan pengguna untuk mengontrol jumlah sumber daya yang tersedia, sementara α , β , dan γ menentukan bagaimana sumber daya tersebut dialokasikan ke kedalaman, lebar, dan resolusi jaringan [15].

2.3.3. Transfer Learning

Proses *train* model CNN dapat dilakukan dengan dua cara yang pertama, membangun jaringan dari awal yaitu membangun arsitektur jaringan dari awal, melatihnya dengan data yang diberikan. yang kedua dengan menggunakan model *pra-trained*, yaitu model yang sebelumnya dilatih untuk tugas yang sama atau serupa untuk dapat diadaptasi pada tugas yang berbeda dengan menyempurnakan lapisan akhir jaringan (*fine-tune*) yang berbeda. [16].

Pada *transfer learning*, *layer* terakhir pada jaringan dapat dihapus atau ditambah dengan *layer* kustom dan dilatih ulang untuk tugas yang berbeda (*fine tune*) [17]. *Transfer learning* memungkinkan model menggunakan pengetahuan dari tugas sebelumnya untuk meningkatkan kinerja pada tugas baru

yang serupa dan dapat mengurangi waktu dan data yang diperlukan untuk melatih model baru serta meningkatkan performa atau akurasi yang lebih tinggi [18].

2.4. Peralatan

Proses perancangan dan pembuatan model pada penelitian ini dilakukan pada mesin AMD-Ryzen 5, 6-Core Processor 4.25 GHz, 16 GB RAM, dengan GPU 2 buah NVIDIA GeForce GTX 1650, menggunakan sistem operasi Windows 10 serta menggunakan dengan initial Desktop Environment Jupyter Notebook dan juga menggunakan Google Collaboratory (colabs).

2.5. Pelatihan

Dalam penelitian ini, digunakan *dataset* berupa citra daun porang yang terdiri dari data asli dan tambahan, yang kemudian dibagi menjadi tiga set yaitu pelatihan, validasi, dan pengujian dengan rasio masing-masing 80%, 10%, dan 10%. Set pelatihan dan validasi digunakan untuk melatih dan menyesuaikan model, sementara set pengujian digunakan untuk mengevaluasi kinerja prediksi pada sampel yang belum pernah dilihat oleh model sebelumnya [19].

Model yang digunakan pada penelitian ini adalah CNN *Custom* dan juga dilakukan *transfer learning*. *Transfer learning* digunakan untuk mempercepat proses pelatihan dengan mengadaptasi model yang telah dilatih pada *dataset ImageNet*, yang terdiri dari sekitar 1,2 juta gambar dan 1000 kategori kelas. Bobot dari model *ImageNet* digunakan untuk mempercepat waktu pelatihan. Pada penelitian ini, model *transfer learning* menggunakan *pre-trained network* yang populer untuk kasus klasifikasi, yaitu VGG16, ResNet50, dan EfficientNetV2M. Penggunaan model *pre-trained* bertujuan untuk mempercepat proses *training* model baru dengan ribuan bahkan jutaan data yang telah di-*train* sebelumnya [20].

Pada proses *training* model, nilai piksel citra pada *dataset* asli dan *augmented* data dinormalisasi dengan membaginya dengan 255 dan kemudian diubah ukurannya menjadi ukuran standar yang diterima oleh masing-masing model. Ukuran gambar yang digunakan adalah 224x224 piksel untuk ResNet50 dan VGG16, serta EfficientNetV2M. *Batch-size* digunakan untuk menentukan ukuran data yang digunakan untuk memperbarui bobot dan bias selama *backpropagation*. Nilai *batch-size* dipilih lebih kecil dari ukuran total *dataset* untuk membagi ukuran total gambar.

Tabel 1. Parameter Ukuran Gambar Pelatihan, Validasi, dan Pengujian

Dataset	Total	Latih (80%)	Validasi (10%)	Uji (10%)
Original	1000	800	100	100
Augmentasi	9000	8100	900	900

Tabel 2. Ukuran Gambar *Input* dan Jumlah Total Parameter Model *Deep Learning*

Model	Ukuran input	Total parameter data
CNN <i>Custom</i>	224 × 224	1000
VGG16	224 × 224	1000
Resnet50	224 × 224	1000
EfficientNetV2M	224 × 224	1000

Tabel 2 menunjukkan jumlah *dataset* dan ukuran gambar input yang digunakan selama proses pelatihan. Parameter ini penting untuk menyeimbangkan kecepatan konvergensi jaringan dan akurasi prediksi. [21].

Proses pelatihan dimulai dengan menginisialisasi beberapa parameter seperti jumlah *epoch* yang berkisar antara 20 hingga 75 *epoch*, *step per epoch* yang disesuaikan dengan jumlah data pengujian dan pelatihan, serta *batch_size* sebesar 32. Pengoptimal yang digunakan adalah Adam, dengan fungsi kerugian (*loss*) *Sparse Categorical Cross Entropy* (SCCE), dan metrik akurasi [22]. Selain itu, data validasi disesuaikan dengan tingkat kebutuhan terhadap penelitian yang dilakukan.

Dalam pelatihan *deep learning*, tingkat kesalahan/kerugian dapat ditentukan menggunakan fungsi *Sparse Categorical Cross Entropy* (SCCE) dan *Categorical Cross Entropy* (CCE). Fungsi ini sering digunakan dalam klasifikasi multikelas, dimana keluaran dari model adalah distribusi probabilitas dari masing-masing kelas. Perbedaan utama antara SCCE dan CCE adalah format label yang digunakan. Dalam SCCE, label diekspresikan dalam format bilangan bulat [0,1], di mana 0 adalah kelas pertama, 1 adalah kelas kedua, dst., sementara di CCE, label diekspresikan sebagai vektor *one-hot encoding* yang memiliki 3 kelas, maka targetnya adalah [1, 0, 0], [0, 1, 0], dan [0, 0, 1]. Tetapi jika dalam bilangan bulat, targetnya menjadi [1, 2, 3]. [23].

Selain itu, *Adaptive Moment Estimation* (Adam), algoritma optimasi yang mengkombinasikan metode *RMSprop* dan SGD dengan Momentum. *RMSprop* digunakan untuk menghitung *adaptive learning rate*, sementara SGD dengan Momentum digunakan untuk menyesuaikan bobot (*weight*) dengan kecepatan yang berbeda tergantung pada gradien. Kombinasi kedua metode ini memungkinkan Adam untuk mempercepat konvergensi dan menghasilkan hasil yang lebih baik pada beberapa tugas *machine learning* [24]. Algoritma ini mudah diimplementasikan, efisien secara komputasi, dan cocok untuk data/parameter yang besar, sehingga banyak digunakan dalam *deep learning* [25].

2.6. Confusion Matrix

Klasifikasi melibatkan pembelajaran berdasarkan data historis melalui serangkaian latihan, dan dalam masalah klasifikasi, pengukuran kinerja umumnya diukur menggunakan *Confusion Matrix* [26] sehingga menghasilkan nilai *precision*, *recall*, dan *akurasi* [27] yang dihitung menggunakan Persamaan 6 hingga Persamaan 9. Untuk menghitung nilai pengukuran tersebut, digunakan indeks *True Positive* (TP): gambar daun patologis (berpenyakit), *True Negative* (TN): gambar daun yang tidak terinfeksi, *False Positive* (FP): gambar daun patologis yang tidak terdeteksi dan *False Negative* (FN): gambar daun yang tidak terinfeksi tetapi salah diklasifikasikan sebagai daun patologis [27].

$$Akurasi = \frac{TruePositive}{TotalData} \dots\dots\dots (6)$$

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive} \dots\dots\dots (7)$$

$$Recall = \frac{TruePositive}{TruePositive + TrueNegative} \dots\dots\dots (8)$$

$$F1\ Score = \frac{2x(Precision \times Recall)}{(Recall + Precision)} \dots\dots\dots (9)$$

3. HASIL DAN PEMBAHASAN

Penelitian ini bertujuan untuk membandingkan akurasi hasil pengujian model CNN *Custom* dengan model *transfer learning* yang telah dilatih dengan jutaan *dataset* gambar dari *Imagenet* dalam mengklasifikasikan hama dan penyakit pada tanaman.

Dataset daun tanaman porang yang digunakan terdiri dari data asli dan telah ditambahkan melalui proses augmentasi, Nilai akurasi, *loss*, dan waktu pelatihan per *epoch* dihitung dan dicantumkan dalam tabel terpisah untuk *dataset* asli dan augmentasi. *Early stop* dan *call back* digunakan selama pelatihan dan nilai terbaik terdapat dalam Tabel 3 hingga Tabel 5 mewakili kasus di mana nilai terbaik diperoleh untuk kriteria performa yang relevan.

Tahap awal dari penelitian ini adalah melatih data awal menggunakan model CNN *Custom* dan model *transfer learning*. Pada tahap ini, beberapa parameter diubah seperti *learning rate*, jumlah *epoch*, dan jumlah lapisan *fully connected*.

Tabel 3. Hasil Ujicoba *Learning Rate*

<i>Learning Rate</i>	Fungsi Aktivasi	Jumlah Percobaan	<i>Minimum Loss</i>	<i>Maximum Loss</i>
0,0001	<i>ReLu</i>	6	0,0002	1,4609
0,001	<i>ReLu</i>	6	0,0004	1,6120

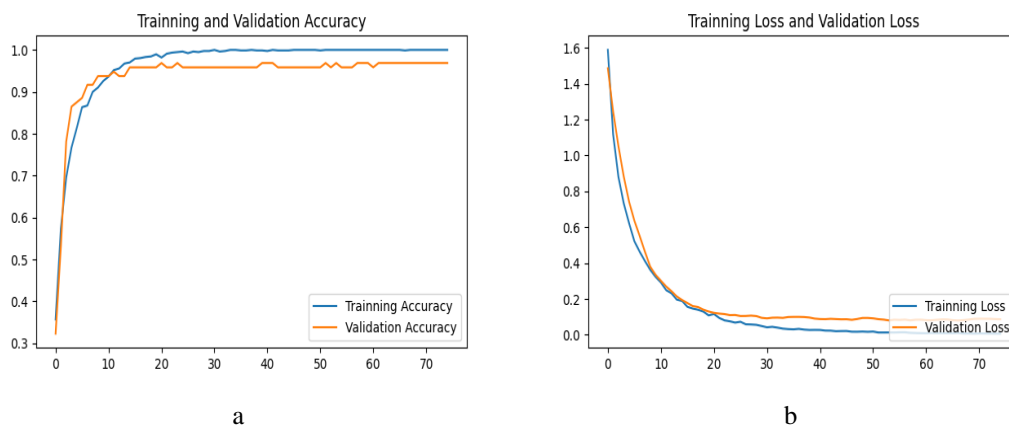
Tabel 4. Hasil Uji Coba *Dataset Original*

Epoch	BASE MODEL											
	<i>CNN Custome</i>			<i>VGG16</i>			<i>ResNet50</i>			<i>EfficientNetV2M</i>		
	<i>Loss</i>	<i>Acc</i>	<i>Time (ms)</i>	<i>Loss</i>	<i>Acc</i>	<i>Time (ms)</i>	<i>Loss</i>	<i>Acc</i>	<i>Time (ms)</i>	<i>Loss</i>	<i>Acc</i>	<i>Time (ms)</i>
20	75,85	78,12	25	15,23	93,75	132	13,18	97,66	114	11,39	97,66	208
28	63,72	85,94	22	11,73	95,31	123	7,58	97,66	101	8,45	96,88	203
36	71,6	88,28	24	12,23	96,09	132	9,90	95,31	110	7,47	97,66	210
44	102,1	89,06	25	10,06	96,88	137	7,41	96,88	109	9,52	96,88	215
50	111,4	87,5	22	15,48	95,31	135	7,56	96,88	113	15,45	97,66	207
75	43,02	84,38	21	21,67	92,71	134	17,68	92,71	118	10,52	97,92	213

Tabel 4. Hasil Uji Coba *Dataset Augmentasi*

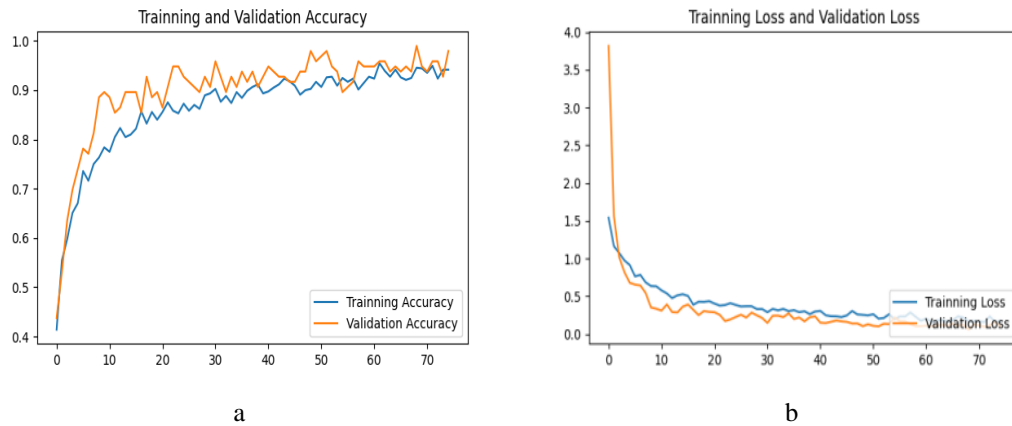
Epoch	BASE MODEL											
	<i>CNN Custome</i>			<i>VGG16</i>			<i>ResNet50</i>			<i>EfficientNetV2M</i>		
	<i>Loss</i>	<i>Acc</i>	<i>Time (ms)</i>	<i>Loss</i>	<i>Acc</i>	<i>Time (ms)</i>	<i>Loss</i>	<i>Acc</i>	<i>Time (ms)</i>	<i>Loss</i>	<i>Acc</i>	<i>Time (ms)</i>
20	113,2	52,3	24	31,1	88,3	136	27,3	91,4	105	23,2	88,3	204
28	98,6	65,6	27	27,1	93,0	138	20,0	95,3	101	24,4	90,6	209
36	74,5	70,3	24	17,8	95,3	127	14,6	95,3	97	21,0	93,0	204
44	67,1	76,6	24	18,4	93,8	129	17,8	92,2	95	15,7	94,5	201
50	62,4	80,5	23	18,9	95,3	133	13,4	94,5	112	10,3	96,9	211
75	55,0	85,9	22	18,7	96,1	140	10,3	97,7	110	7,6	98,4	210

Pada *dataset* asli, semua model memiliki akurasi rata-rata yang sangat dekat satu sama lain seperti yang terlihat di Tabel 4 dan 5. Namun, model *ResNet50* memberikan akurasi rata-rata terbaik untuk semua kelas. Sedangkan keberhasilan rata-rata model dalam memprediksi kelas selain kelas target hampir sama antara model. Presisi atau tingkat klasifikasi yang benar dari sampel yang diklasifikasikan sebagai positif oleh *model ResNet50* lebih tinggi serta waktu *per epoch* pada proses uji (*training*) lebih rendah dari pada model lainnya, yaitu 105 ms.

Gambar 3. *Training and Validation* a). *Accuracy ResNet50*; b). *Loss ResNet50*

Tabel 4 hasil pengujian deteksi dan klasifikasi penyakit serta hama pada tanaman Porang menunjukkan bahwa model *ResNet50* menghasilkan akurasi terbaik. Gambar 3 menunjukkan perbandingan antara akurasi pelatihan dan validasi, serta kehilangan (*loss*) pelatihan dan validasi. Dari grafik tersebut, dapat dilihat bahwa model tidak terlalu mengalami *underfitting* atau *overfitting* pada *dataset* asli.

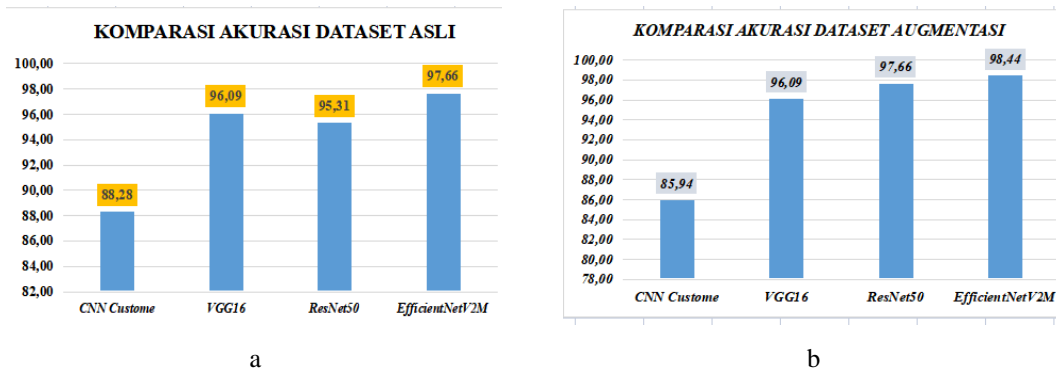
Hasil pengujian menggunakan model *ResNet50* memberikan akurasi tertinggi pada *dataset* asli, sedangkan pada *dataset* yang telah di-augmentasi, model *EfficientNetV2M* memberikan akurasi tertinggi dibandingkan dengan model lainnya. Pengujian dilakukan dengan *batch-size* 32, *learning rate* 0,001, optimasi menggunakan *Adam*, dan aktivasi dengan *softmax*.



Gambar 4. *Training and Validation* a). *Accuracy EfficientNetV2M*; b). *EfficientNetV2M*

Gambar 4 menampilkan akurasi set uji pada *dataset* augmentasi pada model *EfficientNetV2M* mendapatkan akurasi yang cukup baik, untuk deteksi dan klasifikasi penyakit dan hama tanaman porang. perbandingan antara akurasi dan validasi pelatihan, serta kehilangan (*loss*) dan validasi pelatihan sangat kecil sehingga model tidak mengalami *overfitting* atau *underfitting*.

Perbandingan peningkatan akurasi terhadap validasi *set* yang diperoleh pada rangkaian pengujian *dataset* asli dan augmentasi semua model menunjukkan kenaikan yang signifikan. akurasi terhadap data validasi pada Gambar 3, indikasi dihitung sebagai rasio jumlah sampel yang diklasifikasikan dengan benar terhadap semua sampel.



Gambar 5. Uji Akurasi *Deep Learning Models Dataset* a). *Original*; b). *Augmentasi*

Tabel 6. Kinerja Klasifikasi *ResNet50* Pada *Dataset* Asli

	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>	<i>Support</i>
(0)	1,00	1,00	1,00	23
(1)	1,00	0,91	0,95	22
(2)	0,93	1,00	0,96	13
(3)	0,95	1,00	0,97	18
(4)	0,95	0,95	0,95	20
<i>accuracy</i>			0,97	96
<i>macro avg</i>	0,97	0,97	0,97	96
<i>weighted avg</i>	0,97	0,97	0,97	96

Tabel 7. Kinerja Klasifikasi *EfficientNetV2M* Pada *Dataset* Augmentasi

	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>	<i>Support</i>
(0)	1,00	1,00	1,00	23
(1)	1,00	0,91	0,95	22
(2)	0,93	1,00	0,96	13
(3)	0,95	1,00	0,97	18
(4)	0,95	0,95	0,95	20
<i>accuracy</i>			0,97	96
<i>macro avg</i>	0,97	0,97	0,97	96
<i>weighted avg</i>	0,97	0,97	0,97	96

Tabel 6 dan 7 memperlihatkan nilai presisi, *recall*, dan *f1-score* untuk setiap kelas pada dua dataset, yaitu *dataset* asli dan dataset yang telah di-augmentasi menggunakan model *EfficientNetV2M* dan *ResNet50*.

Dari gambar 5 dapat dilihat bahwa model *Efficientnetv2m* memiliki kinerja terbaik pada *dataset* aumentasi, dengan nilai presisi antara 98,44% hingga 100%, sedangkan model *ResNet50* memiliki kinerja terbaik pada dataset asli, dengan nilai presisi antara 97,66% hingga 100%. model *EfficientNetV2M* dan *ResNet50* dianggap berhasil digunakan dalam klasifikasi penyakit daun tanaman.

4. KESIMPULAN

Studi ini mengusulkan penggunaan *Deep Convolutional Neural Network Transfer Learning* untuk klasifikasi hama dan penyakit daun tanaman porang. Dalam penelitian ini, terdapat lima kelas pada *dataset* dan arsitektur yang diusulkan berhasil mencapai akurasi 98,44% pada *dataset* asli dan 97,40% pada *dataset* augmentasi. Teknik *deep learning* telah menjadi populer dalam pemrosesan gambar dan pengenalan pola. Model *EfficientNetV2M* dan *ResNet50* adalah model yang berhasil digunakan dalam deteksi penyakit pada daun tanaman porang.

DAFTAR PUSTAKA

- [1] M. Rofikhoh, Kusnul and Setiahad, Rahmanta and Puspitawati, Indah Rekyani and Lukito, "Potensi produksi tanaman porang (*amorphophallus muelleri* blume) di kelompok tani mpsdh wono lestari desa padas kecamatan dagangan kabupaten madiun.," *J. Agri-Tek*, vol. 17, no. 2, 2017.
- [2] F. Soedarjo, M and Djufry, "Identified diseases would threaten on the expansion of *Amorphophallus muellery* Blume cultivation in Indonesia," *IOP Conf. Ser. Earth Environ. Sci.*, vol. 648, no. 1, p. 012043, 2021.



- [3] M. Brahim, K. Boukhalfa, and A. Moussaoui, "Deep learning for tomato diseases: classification and symptoms visualization," *Appl. Artif. Intell.*, vol. 31, no. 4, pp. 299–315, 2017.
- [4] Y. Lu, Yang and Yi, Shujuan and Zeng, Nianyin and Liu, Yurong and Zhang, "Identification of rice diseases using deep convolutional neural networks," *Neurocomputing*, vol. 267, pp. 378--384, 2017.
- [5] D. S. Fuentes, Alvaro and Yoon, Sook and Kim, Sang Cheol and Park, "A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition," *Sensors*, vol. 17, no. 9, p. 2022, 2017.
- [6] M. Sibiy, Malusi and Sumbwanyambe, "A computational procedure for the recognition and classification of maize leaf diseases out of healthy leaves using convolutional neural networks," *AgriEngineering*, vol. 1, pp. 119--131, 2019.
- [7] X. Xie, Y. Ma, B. Liu, J. He, S. Li, and H. Wang, "A deep-learning-based real-time detector for grape leaf diseases using improved convolutional neural networks," *Front. Plant Sci.*, vol. 11, p. 751, 2020.
- [8] D. Hughes, M. Salathé, and others, "An open access repository of images on plant health to enable the development of mobile disease diagnostics," *arXiv Prepr. arXiv1511.08060*, 2015.
- [9] G. Geetharamani and A. Pandian, "Identification of plant leaf diseases using a nine-layer deep convolutional neural network," *Comput. Electr. Eng.*, vol. 76, pp. 323–338, 2019.
- [10] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv Prepr. arXiv1409.1556*, 2014.
- [11] S. Tammina, "Transfer learning using vgg-16 with deep convolutional neural network for classifying images," *Int. J. Sci. Res. Publ.*, vol. 9, pp. 143--150, 2019.
- [12] J. He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770--778.
- [13] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International conference on machine learning*, 2019, pp. 6105–6114.
- [14] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.
- [15] Y. Altuntaş, Z. Cömert, and A. F. Kocamaz, "Identification of haploid and diploid maize seeds using convolutional neural networks and a transfer learning approach," *Comput. Electron. Agric.*, vol. 163, p. 104874, 2019.
- [16] Y. A. Chen, Junde and Chen, Jinxiu and Zhang, Defu and Sun, Yuandong and Nanekaran, "Using deep transfer learning for image-based plant disease identification," *Comput. Electron. Agric.*, vol. 173, p. 105393, 2020.

- [17] T. Sarkar, Dipanjan and Bali, Raghav and Ghosh, Hands-On Transfer Learning with Python: Implement advanced deep learning and neural network models using TensorFlow and Keras. 2018.
- [18] C. Tan, Chuanqi and Sun, Fuchun and Kong, Tao and Zhang, Wenchang and Yang, Chao and Liu, "A survey on deep transfer learning," *Int. Conf. Artif. neural networks*, pp. 270--279, 2018.
- [19] A. J. Li, Mu and Zhang, Tong and Chen, Yuqiang and Smola, "Efficient mini-batch training for stochastic optimization," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2014, pp. 661--670.
- [20] M. Zhang, Zhilu and Sabuncu, "Generalized cross entropy loss for training deep neural networks with noisy labels," *Adv. Neural Inf. Process. Syst.*, vol. 31, 2018.
- [21] J. Klein, "Understanding the Difference between Python Sparse Categorical Cross-Entropy and Categorical Cross-Entropy Loss Functions," 2023.
- [22] S. Gulli, Antonio and Pal, Deep learning with Keras. Packt Publishing Ltd, 2017.
- [23] V. Bushaev, "Adam—latest trends in deep learning optimization," *Towar. Data Sci. List.*, 2018.
- [24] J. Kingma, Diederik P and Ba, "Adam: A method for stochastic optimization," *arXiv Prepr. arXiv1412.6980*, 2014.
- [25] J.-J. and others Sheu, "An Efficient Two-phase Spam Filtering Method Based on E-mails Categorization.," *Int. J. Netw. Secur.*, vol. 9, pp. 34--43, 2009.
- [26] A. Kurniasari, D. Erwanto, dan P. N. Rahayu, "Ekstraksi Fitur Tekstur dan Warna pada Kulit Katak Menggunakan GLCM dan Momen Warna," *Jurnal ELTIKOM*, vol. 6, no. 1, 2022, doi: 10.31961/eltikom.v6i1.287
- [27] R. Arthana, "Mengenal Accuracy, Precision, Recall dan Specificity serta yang diprioritaskan dalam Machine learning," *Mediu. com*, 2019