Deteksi Penyakit Tanaman Paprika, Tomat, dan Kentang dengan menggunakan *InceptionV3*

Fauzan Cahyadi
Teknik Telekomunikasi
Fakultas Teknik Elektro
Bandung, Indonesia
fauzancahyadi@student.telkomunversity.ac.id

Syamsul Rizal
Teknik Telekomunikasi
Fakultas Teknik Elektro
Bandung, Indonesia
syamsulrizal@telkomunversity.ac.id

Sofia Sa'idah
Teknik Telekomunikasi
Fakultas Teknik Elektro
Bandung, Indonesia
sofiasaidahsfi@telkomunversity.ac.id

***Abstrak* — Penyakit tanaman menjadi masalah karena dapat menyebabkan penurunan kualitas dan kuantitas produk pertanian serta peningkatan biaya total kerusakan. Penyakit tanaman bisa diketahui dari kondisi daun pada tanaman. Pada zaman *modern* ini telah terdapat penelitian yang dapat mengklasifikasi penyakit dengan menggunakan citra daun tanaman. Metode tersebut yaitu dengan menggunakan *Convolutional Neural Network* (CNN). Pada Jurnal ini mengusulkan *InceptionV3* sebagai arsitektur CNN. *InceptionV3* adalah arsitektur CNN yang dikemukakan oleh Google yang bisa melatih data sampai 1000 kelas dan lebih dari 1.4 juta citra. *Dataset* yang digunakan pada penelitian ini bersumber dari kaggle.com dan *PlantVilagge*. *Dataset* berisi 10800 citra daun dengan format .jpeg. Pengklasifikasi pada citra dibagi menjadi 9 kelas yaitu 2 kelas pada paprika, 3 kelas pada kentang, dan 4 kelas pada tomat. Parameter yang digunakan adalah akurasi, sensitivitas, presisi, dan *f1score*. Hasil pengujian terbaik didapatkan menggunakan *optimizer* SGD, *learning rate* 0.001, *batch size* 32, *epoch* 100 memperoleh tingkat akurasi 91% dan *loss* 0.2568, presisi 91%, sensitivitas(*recall*) 91%, dan *F1score* 91%.**

***Kata kunci*— Penyakit tanaman, InceptionV3, CNN.**

I. PENDAHULUAN

Pertanian dan perkebunan merupakan cara untuk mengembangkan sumber hayati atau tanaman dalam upaya untuk memenuhi kebutuhan hidup sehari-hari. Sumber hayati atau tanaman seperti paprika, kentang, dan tomat memiliki banyak manfaat bagi tubuh. Salah satu manfaat paprika dan tomat dapat meningkatkan dan menjaga kesehatan mata. Kentang bisa sebagai pengganti sumber karbohidrat. Untuk memaksimalkan manfaatnya tentu diperlukan hasil panen yang bagus dan sehat. Penyakit tanaman tentu menjadi masalah karena dapat menyebabkan penurunan kualitas dan kuantitas produk pertanian serta peningkatan biaya total kerusakan [1]. Penyakit pada tanaman bisa diketahui dengan melakukan analisa pada bagian-bagian dari tanaman seperti daun, akar, dan batang.

Pada zaman modern ini telah terdapat penelitian yang dapat menganalisa penyakit pada tanaman. Metode tersebut yaitu dengan menggunakan *Convolutional Neural Network* (CNN). CNN adalah sebuah tipe deep learning yang digunakan untuk proses pengolahan data yang mempunyai pola *grid*, seperti citra [2]. Citra yang diolah yaitu citra daun yang sehat dan citra daun yang menunjukkan gejala penyakit pada tanaman. CNN akan mengolah citra daun yang kemudian citra tersebut akan diklasifikasikan berdasarkan penyakit pada tanaman.

CNN mempunyai jenis-jenis arsitektur seperti VGG, *ResNet, Inception,* dan *GoogleNet*. Konstantinos P. Ferentinos melakukan training pada database 87848 citra tanaman dimana terdapat 58 kelas berbeda (tanaman dan penyakit), juga termasuk tanaman yang sehat [3]. Ferentinos menggunakan arsitektur *AlexNet, AlexNetOWTBn, GoogLeNet, Overfeat*, and VGG. Hasil penelitiannya menunjukkan *successs rate* tertinggi didapatkan dengan VGG yaitu 99.53% dan tingkat *average error* terendah didapatkan dengan model *AlexNetOWTBn* yaitu 0.0192 [3]. Lilian Mkonyi et al. mengusulkan model deep learning yaitu VGG16, VGG19, dan *ResNet50* untuk mengidentifikasi *T. absoluta* pada tanaman tomat. Hasil dari penelitiannya menunjukkan performa terbaik pada VGG16 yang mencapai akurasi keseluruhan 91.9% [4]. Parul Sharma et al. meneliti deteksi penyakit pada tanaman menggunakan S-CNN yang berbasis *image segmentation* dengan akurasi 98,6% [5]. Vinod Kumar et al. meneliti deteksi penyakit pada daun tanaman dengan menggunakan *ResNet34* yang menghasilkan akurasi 99.40% [6]. Ishrat Zahan Mukti dan Dipayan Biswas mengusulkan *transfer training* untuk mengidentifikasikan penyakit tanaman menggunakan *ResNet50*. Penelitian mereka menghasilkan tingkat akurasi 99.80% [7].

Pada penelitian ini, penulis akan menggunakan CNN dengan arsitektur *InceptionV3* untuk mendeteksi penyakit pada tanaman. *InceptionV3* merupakan versi ketiga dari *Inception* atau *GoogleNet*. *InceptionV3* dikemukakan oleh *Google* yang bisa melatih data sampai 1000 kelas dan lebih dari 1.4 juta citra [8]. Parameter yang akan digunakan pada

penelitian ini untuk mengukur tingkat performa yaitu tingkat akurasi, sensitivitas, presisi, dan *F1score*.

Penelitian ini bertujuan merancang dan mengimplementasikan *Convolutional Neural Network* (CNN) dengan arsitektur *InceptionV3* untuk mendeteksi penyakit pada tanaman. Tujuan penelitian ini juga mencakup analisis performa dari hasil penelitian berdasarkan tingkat akurasi dan waktu komputasi yang dibutuhkan untuk mendeteksi penyakit tanaman.

II. KAJIAN TEORI

A. Tanaman

Keberadaan tanaman sangat mempengaruh keadaan ekosistem dan sumber pangan. Penyakit tentu akan berdampak buruk pada tanaman dan lahan pertanian. Penyebab utama penyakit ini adalah mikroorganisme, kelainan genetik, dan agen infeksi seperti bakteri, jamur, dan virus [9]. Pada penelitian ini, penulis akan menggunakan citra daun pada tanaman paprika, kentang, dan tomat.

Paprika mengandung antikoagulan yang membantu mencegah pembekuan darah yang dapat menyebabkan serangan jantung dan juga merupakan sumber vitamin C [10]. Kentang memiliki manfaat untuk mengontrol tekanan darah, untuk diet sehat, mengurangi resiko penyakit kronis, mengontrol kadar gula dalam darah, memelihara fungsi otak dan saraf, dan memperlancar pencernaan. Selain itu, kentang juga mengandung vitamin C dan B6, kalium, magnesium, dan zat besi[11]. Tomat memiliki manfaat untuk menjaga kesehatan jantung, mencegah kanker, melancarkan pencernaan, menjaga kesehatan mata, dan menjaga kesehatan kulit.

B. Citra

Citra adalah representasi dari sebuah objek, seseorang, atau sebuah kejadian[12]. Citra digital yaitu fungsi dua dimensi f(x,y) yang memproyeksikan pemandangan tiga dimensi menjadi bentuk dua dimensi, dimana x, y mempresentasikan lokasi sebuah objek atau *pixel* dan memiliki nilai intensitas[12]. *Pixel* merupakan bagian terkecil dari citra yang memuat informasi dari citra dan dinyatakan dengan bilangan bulat. Nilai intensitas citra biasanya dinyatakan dalam kisaran *range* [0, 255]. Citra RGB (*Red, Green, Blue*) adalah citra yang nilai intensitas (*range*) pikselnya tersusun oleh tiga warna dasar, yaitu merah, hijau, dan biru. Sementara itu, citra *grayscale* adalah citra yang nilai intensitas pikselnya didasarkan pada intensitas (*range*) keabuan. Sedangkan citra biner adalah citra yang hanya memiliki dua nilai intensitas, yaitu 0 (hitam) dan 1 (putih).

C. *Confusion Matrix*

Confusion Matrix pada dasarnya memberikan informasi tentang perbandingan hasil klasifikasi yang dilakukan oleh sistem (model) dengan hasil klasifikasi sebenarnya. *Confusion Matrix* berbentuk tabel matriks yang menggambarkan kinerja model klasifikasi pada serangkaian data uji di mana nilai sebenarnya diketahui. Terdapat 4 istilah yang digunakan sebagai representasi hasil proses klasifikasi pada *Confusion Matrix*, yang terdapat pada Gambar 1, yaitu *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN). *Confusion Matrix* digunakan untuk menghitung nilai akurasi, presisi, sensitivitas (*recall*), dan *F1-score*.

		Prediksi	
		0 (Negatif)	1 (Positif)
Aktual	0 (Negatif)	T_N	F_P
	1 (Positif)	F_N	T_P

Gambar 1. Confusion Matrix

D. *Convolutional Neural Network* (CNN)

CNN adalah sebuah tipe deep learning yang untuk proses pengolahan data yang mempunyai pola grid, seperti citra [2]. CNN merupakan evolusi dari *Artificial Neural Network* (ANN) yang berfokus pada *image processing* [3, 6]. Hal ini disebabkan karena CNN menggunakan lebih sedikit parameter daripada ANN. CNN bertujuan untuk mengurangi ukuran citra tanpa menghilangkan fitur-fitur penting pada citra. Dengan itu CNN bisa mendeteksi dan mengenali objek pada citra sehingga memudahkan untuk mengklasifikasikan citra berdasarkan jenis objek. *Convolutional Neural Network* (CNN) memiliki beberapa layer, antara lain *Convolutional Layer, Pooling Layer*, dan *Fully-connected Layer*.

Convolutional Layer merupakan salah satu *layer* pada CNN yang berfungsi untuk mengurangi ukuran citra dengan menerapkan konvolusi menggunakan matriks *filter* [6]. *Pooling Layer* merupakan *layer* yang mengurangi ukuran citra dan mengekstrak ciri-ciri dominan pada citra atau gambar. Pada layer ini, diterapkan fungsi seperti *Max-pooling* dan *Avg-pooling* [6]. *Max-pooling* dilakukan dengan memilih nilai terbesar di dalam sub jendela dan nilai tersebut ditransfer ke dalam matriks baru [13]. Sementara itu, *Avg-pooling* dilakukan dengan mengambil nilai rata-rata dari sub jendela dan nilai rata-rata tersebut ditransfer ke dalam matriks baru.

Fully-connected Layer merupakan layer yang bertugas melakukan pengenalan (*recognition*) dan klasifikasi [13]. Input pada layer ini berupa matriks yang dihasilkan dari *convolutional layer* dan *pooling layer* [6, 13]. Semua layer terhubung ke semua layer sebelumnya dan layer selanjutnya melalui konektivitas tepi antara *neuron* yang ada di dalamnya [6, 14]. *Fully-connected layer* hanya digunakan jika ukuran citra telah cukup dikurangi oleh rangkaian *convolutional layer plus pooling* sehingga lapisan yang terhubung sepenuhnya tidak memiliki banyak parameter untuk dipelajari [6].

E. *InceptionV3*

InceptionV3 adalah arsitektur CNN yang dikemukakan oleh Google yang bisa melatih data sampai 1000 kelas dan lebih dari 1.4 juta citra [8]. *Inception network* atau *GoogLeNet* memiliki beberapa versi seperti *InceptionV1, InceptionV2, InceptionV3, InceptionV4, Inception-Resnet. InceptionV3* memiliki empat blok atau iterasi yang memiliki layer atau proses seperti konvolusi, *avgpooling*, dan *maxpooling* yang memiliki keluaran *concat* [15].

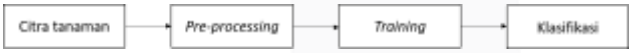
F. *MobileNetV2*

MobileNetV2 merupakan arsitektur model CNN yang ringan dan dapat diimplementasikan pada perangkat *mobile*. *MobileNetV2* adalah hasil perkembangan dari model sebelumnya, yaitu *MobileNet*. Tujuan dibentuknya *MobileNet* adalah untuk memberikan kinerja yang tinggi dengan menggunakan *hyperparameter* yang membuat model menjadi lebih kecil dan mempercepat proses perhitungan[16]. *MobileNet* mempunyai lapisan inti yang disebut sebagai *Depthwise Separable Convolutions* yang memecah atau membagi konvolusi menjadi dua layer, yaitu *depthwise convolution* dan *pointwise convolution* [16,17]. Seiring perkembangannya, *MobileNetV2* menghadirkan *Linear Bottlenecks* dan *Inverted Residual* yang dapat meningkatkan performa *MobileNet* itu sendiri [18].

III. METODE

A. Desain Sistem

Pada penelitian ini, *Convolutional Neural Network* (CNN) digunakan untuk mendeteksi penyakit pada tanaman seperti paprika, tomat, dan kentang. Input yang digunakan adalah citra daun tanaman yang dapat berupa citra dari tanaman yang sehat maupun yang terkena penyakit. Citra daun tanaman ini didapatkan dari situs *Kaggle* dan merupakan milik PlantVillage. Jumlah citra daun yang digunakan untuk dataset adalah 10.800 citra dengan format .jpeg. Metode yang digunakan dalam penelitian ini adalah menggunakan *Convolutional Neural Network* (CNN) dengan arsitektur *InceptionV3*. Sebelum citra input diproses oleh CNN *InceptionV3*, citra-citra tersebut harus melalui proses *pre-processing* terlebih dahulu. Pada Gambar 2. menjelaskan proses dari sistem yang dirancang.



Gambar 2. Desain Sistem

B. *Dataset*

Dataset yang digunakan dalam penelitian ini terdiri dari citra daun tanaman paprika, citra daun tanaman kentang, dan citra daun tanaman tomat. Setiap kelas memiliki jumlah citra sebanyak 1200, sehingga total jumlah citra dalam dataset adalah 10800 citra. Dataset ini terbagi menjadi 9 kelas berdasarkan jenis dan kondisi daun tanaman yang berbeda. Pada Tabel 1. menjelaskan rincian *dataset* dari penelitian ini.

Tabel 1 Rincian Dataset

Tanaman	Penyakit	Jumlah Data
Paprika	*Bacterial spot*	1200
Paprika	*Healty*	1200
Kentang	*Early blight*	1200
Kentang	*Healty*	1200
Kentang	*Late blight*	1200
Tomat	*Target spot*	1200
Tomat	*Healty*	1200
Tomat	*Leaf mold*	1200
Tomat	*Septoria leaf spot*	1200
Total		10800

C. *Pre-processing*

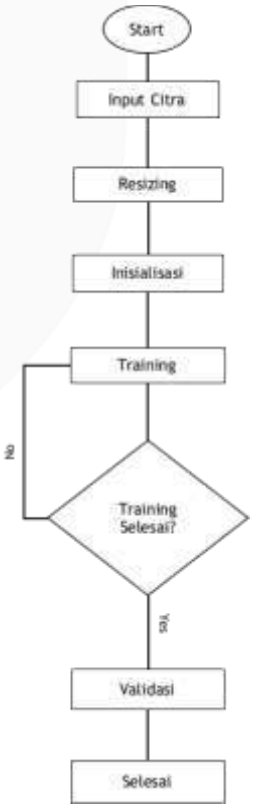
Pada tahap ini, *pre-processing* dilakukan untuk memperbaiki kualitas citra dan menyeragamkan jumlah citra dalam *dataset*. *Dataset* pada penelitian ini terdiri dari citra RGB dengan ukuran awal 256x256 piksel. *Pre-processing* yang digunakan meliputi dua tahap, yaitu augmentasi dan *resize*.

Proses Augmentasi dilakukan untuk menyeimbangkan jumlah citra dalam setiap kelasnya. Hal ini dilakukan dengan menggandakan citra yang jumlahnya kurang dalam suatu kelas, menggunakan citra yang sudah ada, dan menerapkan teknik rotasi dan refleksi untuk mendapatkan variasi data yang lebih banyak.

Setelah proses Augmentasi selesai, jumlah data setiap kelas mencapai 1200 citra dengan variasi rotasi dan refleksi yang telah diterapkan. Selanjutnya, dilakukan proses *resize* untuk mengubah ukuran citra menjadi 128x128 piksel. Dengan demikian, *dataset* telah di-preprocess dan siap untuk digunakan dalam pelatihan model *Convolutional Neural Network* (CNN) untuk mendeteksi penyakit pada tanaman.

D. *Training*

Pada tahap ini, citra yang telah melalui *pre-processing* akan diproses melalui *Convolutional Neural Network* (CNN) agar dapat diklasifikasikan berdasarkan kelas-kelas citra tersebut. Sebelum memulai proses pelatihan (*training*) pada CNN, terlebih dahulu dilakukan proses inisialisasi seperti pada Gambar 3. di mana langkah-langkah diambil untuk menetapkan nilai *hyperparameter*. *Hyperparameter* ini mencakup nilai *optimizer, learning rate, batch size*, dan jumlah *epoch* yang akan digunakan selama proses pelatihan model. Dengan nilai hyperparameter yang tepat, diharapkan model dapat mengoptimalkan proses pelatihan dan memberikan hasil klasifikasi yang akurat.



Gambar 3. Flowchart Proses Training dan Testing

E. Parameter Performa

Setelah melakukan eksekusi pada sistem untuk mengetahui hasil dari penelitian, diperlukan parameter performa untuk mengevaluasi kinerja dari model yang telah dilatih. Pada penelitian ini, parameter performa yang akan digunakan sebagai berikut:

1. Akurasi

Akurasi adalah perhitungan yang membandingkan rasio jumlah prediksi yang benar dengan total prediksi[6]. Berikut adalah persamaan untuk menghitung akurasi:

$$Akurasi = \left(\frac{T_P + T_N}{T_P + T_N + F_P + F_N}\right) 100\% \qquad (1)$$

2. Sensitivitas

Sensitivitas adalah perhitungan yang dilakukan untuk mengetahui seberapa sering sistem mendeteksi data yang benar [19]. Berikut adalah persamaan untuk menghitung sensistivitas:

$$Sensitivitas = \left(\frac{T_P}{T_P + F_N}\right) 100\% \qquad (2)$$

3. Presisi

Presisi adalah perbandingan jumlah positif benar dengan jumlah total prediksi positif [6]. Berikut adalah persamaan untuk menghitung presisi:

$$Presisi = \left(\frac{T_P}{T_P + F_P}\right) 100\% \qquad (3)$$

4. *F1score*

F1score adalah perhitungan melibatkan presisi dan recall relatif terhadap kelas positif tertentu. Ini dapat diartikan sebagai *weighted average* dari presisi dan *recall*, dimana *F1score* mencapai nilai terbaiknya pada 1 dan terburuk pada 0[6]. Berikut adalah persamaan untuk menghitung *F1score*:

$$F1score = \left(\frac{2(Presisi \times Sensitivitas)}{Presisi + Sensitivitas}\right) 100\% \qquad (4)$$

IV. HASIL DAN PEMBAHASAN

Penelitian akan diuji menggunakan lima skenario yaitu perbandingan *optimizer*, perbandingan *learning rate*, perbandingan *batch size*, perbandingan *epoch*, perbandingan hasil *InceptionV3* dan *MobileNetV2*.

Pada penelitian ini, dilakukan pengujian menggunakan berbagai jenis *optimizer*, yaitu *Adam*, SGD, *Adamax, Nadam*, dan *RMSprop*. Pengujian ini bertujuan untuk menemukan *optimizer* yang memberikan hasil yang lebih baik dalam mendeteksi penyakit pada tanaman. Pengujian dilakukan dengan mengambil hasil terbaik dari pengujian *optimizer* dengan mengubah nilai *learning rate*. *Learning rate* yang digunakan dalam pengujian ini adalah 0.0001, 0.001, dan 0.01.

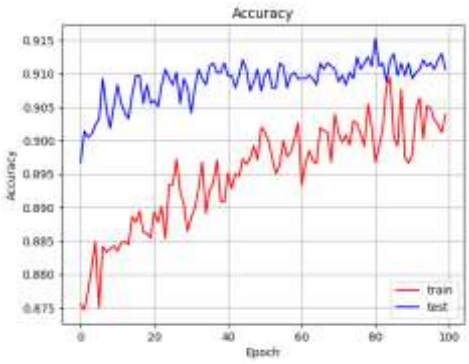
Selanjutnya, pengujian dilakukan dengan mengambil parameter terbaik dari dua pengujian sebelumnya, yaitu *optimizer* SGD dan *learning rate* 0.001. Dalam pengujian ini, nilai *batch size* diubah menjadi 8, 16, 24, dan 32. Parameter lain yang digunakan adalah *epoch* 100.

Selanjutnya, dilakukan pengujian dengan mengubah nilai *epoch*, yaitu 20, 40, 60, 80, 100, dan 120. Parameter yang digunakan dalam pengujian ini adalah *optimizer* SGD, *learning rate* 0.001, dan *batch size* 32. Dengan cara ini, diharapkan dapat menemukan kombinasi *hyperparameter* yang optimal untuk model dalam mendeteksi penyakit pada tanaman.

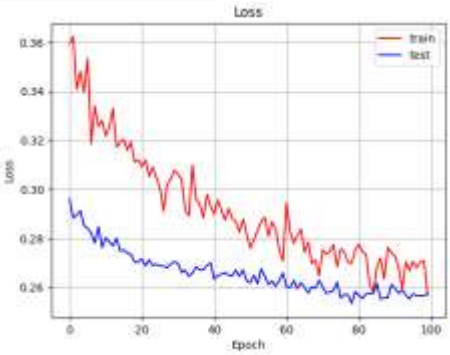
Setelah melakukan pengujian pada empat parameter, yaitu *optimizer, learning rate, batch size*, dan *epoch*, didapatkan hasil dapat dilihat pada Tabel 2.

Tabel 2 Rincian Hasil Penelitian

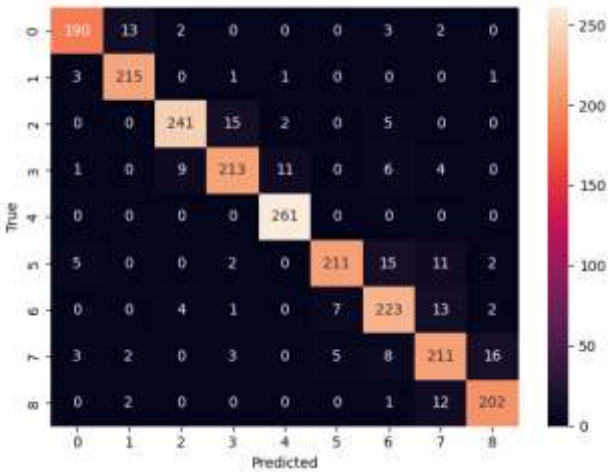
Parameter	**Nilai**
Optimizer	SGD
Learning Rate	0.001
Batch Size	32
Epoch	100
Akurasi	91%
Loss	0.2568
Presisi	91%
Sensitivitas (*Recall*)	91%
F1score	91%



Gambar 4. Grafik Akurasi Hasil Penelitian



Gambar 5. Grafik Loss Hasil Penelitian



Gambar 6. Confusion Matrix Hasil Penelitian

Confusion Matrix pada penelitian ini terdiri dari 9 kelas yang mencakup kelas 0 sampai 8. Setiap kelas ini mewakili jenis penyakit atau keadaan kesehatan pada tanaman tertentu, seperti *Pepper Bacterial Spot, Pepper Healthy, Potato Early blight, Potato Healthy, Potato Late blight, Tomato Leaf Mold, Tomato Septoria Leaf Spot, Tomato Target Spot,* dan *Tomato Healthy*. Pada Gambar 6. menunjukkan hasil *Confusion Matrix* yang menggambarkan kemampuan sistem dalam memprediksi data sesuai dengan data aktual untuk setiap kelas penyakit pada tanaman. Dengan hasil *Confusion Matrix* ini, dapat dievaluasi sejauh mana kinerja model dalam mengklasifikasikan citra daun tanaman dan mengidentifikasi penyakit yang ada pada setiap tanaman dengan akurat.

Tabel 3 Perbandingan InceptionV3 dan MobileNetV2

Parameter	***InceptionV3***	***MobileNetV2***
Akurasi	91%	92%
Loss	0.2568	0.0166
Presisi	91%	92%
Sensitivitas	91%	92%
F1score	91%	92%

Tahap selanjutnya adalah melakukan tambahan pengujian untuk membandingkan hasil penelitian menggunakan arsitektur CNN *InceptionV3* dan *MobileNetV2* yang hasilnya terdapat pada Tabel 3. Dalam pengujian ini, digunakan parameter *optimizer* SGD, *learning rate* 0.001, *batch size* 32, dan *epoch* 100.

Hasil pengujian menunjukkan bahwa model yang menggunakan arsitektur *MobileNetV2* memiliki kinerja yang lebih baik dibandingkan dengan model yang menggunakan arsitektur *InceptionV3*. Model dengan arsitektur *MobileNetV2* mencapai tingkat akurasi sebesar 92% dengan nilai *loss* sebesar 0.0166. Dalam hal klasifikasi penyakit pada tanaman, *MobileNetV2* menunjukkan hasil yang lebih unggul daripada *InceptionV3* berdasarkan hasil evaluasi akurasi dan loss.

Perbedaan ini muncul karena terdapat perbedaan dalam proses dan blok yang digunakan pada *InceptionV3* dan *MobileNetV2*. Pada *MobileNetV2*, terdapat penggunaan blok khusus seperti *Depthwise Convolution* yang berkontribusi pada peningkatan performa dan pengurangan nilai loss dari model tersebut.

V. KESIMPULAN

Pada penelitian ini, penulis merancang sebuah sistem yang bertujuan untuk mengklasifikasikan citra daun pada tanaman paprika, kentang, dan tomat menggunakan *Convolutional Neural Network* (CNN) dengan arsitektur *InceptionV3*. Sistem ini memiliki dua kesimpulan utama. Pertama, sistem yang dirancang berhasil mengklasifikasikan sembilan jenis penyakit pada daun, termasuk dua jenis penyakit pada tanaman paprika, tiga jenis penyakit pada tanaman kentang, dan empat jenis penyakit pada tanaman tomat. Hal ini menunjukkan bahwa model CNN *InceptionV3* mampu dengan baik dalam mengidentifikasi berbagai jenis penyakit pada daun dari berbagai tanaman. Kedua, setelah melaksanakan empat pengujian dengan berbagai konfigurasi, hasil terbaik diperoleh dengan menggunakan *optimizer* SGD, *learning rate* 0.001, *batch size* 32, dan *epoch* 100, dengan akurasi mencapai 91% dan *loss* sebesar 0.2568. Performa sistem secara keseluruhan dalam penelitian ini mencapai nilai rata-rata 91%, menunjukkan konsistensi dan kehandalan dalam pengklasifikasian citra daun. Hasil ini memberikan indikasi positif bahwa penggunaan CNN arsitektur *InceptionV3* menjadi solusi yang efektif dalam mendeteksi penyakit pada tanaman secara akurat dan efisien.

Berdasarkan hasil penelitian ini, terdapat beberapa saran yang dapat diberikan kepada penulis selanjutnya untuk pengembangan lebih lanjut. Pertama, disarankan untuk menggunakan jenis citra daun tanaman yang lebih terfokus, misalnya hanya satu atau dua jenis tanaman, sehingga kelas yang harus diidentifikasi lebih sedikit. Dengan demikian, fokus pada setiap jenis tanaman dapat ditingkatkan, dan performa sistem dapat lebih optimal dalam mengklasifikasikan penyakit pada daun tersebut.

Kedua, penulis selanjutnya dapat melakukan eksperimen dengan menggunakan arsitektur CNN yang berbeda selain *InceptionV3*. Misalnya, penggunaan arsitektur seperti *ResNet, DenseNet,* atau *EfficientNet*, yang memiliki struktur yang berbeda dan karakteristik yang unik, dapat memberikan wawasan baru tentang kinerja sistem dalam mengklasifikasikan citra daun.

Dengan mempertimbangkan saran-saran tersebut, penelitian selanjutnya diharapkan dapat menghasilkan perbaikan dan pengembangan lebih lanjut pada sistem pengklasifikasi citra daun tanaman, serta memberikan kontribusi positif dalam bidang deteksi penyakit pada tanaman melalui pemanfaatan teknologi *Convolutional Neural Network* (CNN).

REFERENSI

[1]M. Bhagat, R. Mahmood, M. Kumar, D. Kumar, and B. Pati, “Bell pepper leaf disease classification using cnn,” 2nd International Conference on Data, Engineering and Applications (IDEA), 2020.

[2] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, “Convolutional neural networks: an overview and application in radiology,” Insights into Imaging, vol. 9, no. 4, 2018.

[3] K. P. Ferentinos, "Deep learning models for plant disease detection and diagnosis," Computers and Electronics in Agriculture, vol. 145, no. 311-318, 2018.

[4] L. Mkonyi, D. Rubanga, M. Richardc, N. Zekeyaa, S. Sawahiko, B. Maiseli, and D. Machuvea, "Early identification of tuta absoluta in tomato plants using deep learning," Scientific African, vol. 10, 2020.

[5] P. Sharma, Y. P. S. Berwal, and W. Ghai, "Performance analysis of deep learning cnn models for disease detection in plants using image segmentation," Information Processing in Agriculture, no. xxxx, 2019.

[6] V. Kumar, H. Arora, Harsh, and J. Sisodia, "Resnet-based approach for detection and classification of plant leaf diseases," Proceedings of the International Conference on Electronics and Sustainable Communication Systems, ICESC 2020, no. Icesc, 2020.

[7] I. Z. Mukti and D. Biswas, "Transfer learning based plant diseases detection using resnet50," 2019 4th International Conference on Electrical Information and Communication Technology, EICT 2019, no. December, 2019.

[8] K. Joshi, V. Tripathi, C. Bose, and C. Bhardwaj, "Robust sports image classification using inceptionv3 and neural networks," Procedia Computer Science, vol. 167, 2020.

[9] D. Tiwari, M. Ashish, N. Gangwar, A. Sharma, S. Patel, and D. S. Bhardwaj, "Potato leaf diseases detection using deep learning," Proceedings of the International Conference on Intelligent Computing and Control Systems, ICICCS 2020, no. Iciccs, 2020.

[10] M. Nadeem, F. M. Anjum, M. R. Khan, M. Saeed, and A. Riaz, "Antioxidant potential of bell pepper (capsicum annum l .) -a review," vol.21, no. 1, 2011.

[11] R. A. Sholihati, I. A. Sulistijono, A. Risnumawan, and E. Kusumawati, "Potato leaf disease classification using deep learning approach," in 2020 International Electronics Symposium (IES), 2020.

[12] V. Tyagi, "Understanding digital image processing," Understanding Digital Image Processing, no. September, 2018.

[13] M. Sardogan, A. Tuncer, and Y. Ozen, "Plant leaf disease detection and classification on cnn with lvq algorithm," 2018 3rd International Conference on Computer Science and Engineering (UBMK), 2018.

[14] S. Hari, M. M.Sivakumar, D. P.Renuga, S.karthikeyan, and S.Suriya, "Detection of plant disease by leaf image using convolutional neural network," in 2019 International Conference on Vision Towards Emerging Trends in Communication and Networking (ViTECoN), 2019.

[15] C. Wang, D. Chen, L. Hao, X. Liu, Y. Zeng, J. Chen, and G. Zhang, "Pulmonary image classification based on inception-v3 transfer learning model," IEEE Access, vol. 7, 2019.

[16] X. Xu, M. Du, H. Guo, J. Chang, and X. Zhao, "Lightweight facenet based on mobilenet," International Journal of Intelligence Science, 2021, 11, 1-16, 2020.

[17] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," 2017.

[18] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018.

[19] L. Boukhris, J. B. Abderrazak, and H. Besbes, "Tailored deep learning based architecture for smart agriculture," 2020 International Wireless Communications and Mobile Computing, IWCMC 2020, no. 1, 2020.