

Terakreditasi SINTA Peringkat 4

Surat Keputusan Dirjen Penguatan Riset dan Pengembangan Ristek Dikti No. 28/E/KPT/2019
masa berlaku mulai Vol.3 No. 1 tahun 2018 s.d Vol. 7 No. 1 tahun 2022

Terbit online pada laman web jurnal:
<http://publishing-widyagama.ac.id/ejournal-v2/index.php/jointecs>



Vol. 5 No. 1 (2020) 09 - 16

JOINTECS

(Journal of Information Technology and Computer Science)

e-ISSN:2541-6448

p-ISSN:2541-3619

Aplikasi Pendeteksi Penyakit Pada Daun Tanaman Apel Dengan Metode Convolutional Neural Network

Guntur Wicaksono¹, Septi Andryana², Benrahman³

Program Studi Teknik Informatika, Fakultas Teknologi Komunikasi dan Informatika, Universitas Nasional

¹gunturw79@gmail.com, ²septi.andryana@civitas.unas.ac.id, ³benrahman@civitas.unas.ac.id

Abstract

According to 2017 statistical fruit and vegetable crops published by BPS, total apple production in 2017 amounted to 319004 tons. There are many diseases that can attack apple plants, therefore early detection and identification of plant diseases are the main factors to prevent and reduce the spread of apple plant diseases. CNN method is used in this study with LeNet-5 architecture which can process 3151 imagery data with a mini-mum accuracy level of 75%. This study uses a dataset derived from PlantVillage created by SP Mohanty CEO & Co-founder of CrowdAI with a total of 3151 leaf images that have been classified according to their respective classes. CNN stages include Convolution Layer, Rectified Linear Unit (ReLU), Subsampling, Flattening, Fully Connected Layer. The test results are evaluated using image testing data. The evaluation process is done using a confusion matrix. Based on the results of testing applications that are designed with 99,4% model accuracy and 97,8% validation accuracy, the application is useful for detecting apple disease using apple leaf images.

Keywords: apple leaf disease; convolutional neural networks; image classification; LeNet-5

Abstrak

Menurut data statistik tanaman buah dan sayuran tahun 2017 yang dipublikasikan oleh BPS, jumlah total produksi apel pada tahun 2017 sebesar 319004 ton. Terdapat banyak penyakit yang dapat menyerang tanaman apel, oleh karena itu pendeteksian dini serta pengidentifikasian penyakit tanaman menjadi faktor utama untuk mencegah dan mengurangi penyebaran penyakit tanaman apel. Metode CNN digunakan pada penelitian ini dengan arsitektur LeNet-5 yang dapat mengolah 3151 data citra dengan batas tingkat akurasi minimum sebesar 75%. Penelitian ini menggunakan dataset yang berasal dari PlantVillage yang dibuat oleh SP Mohanty CEO & Co-founder CrowdAI dengan jumlah sebanyak 3151 citra daun yang telah diklasifikasi berdasarkan kelasnya masing-masing. Tahapan CNN diantaranya Convolution Layer, Rectified Linear Unit (ReLU), Subsampling, Flattening, Fully Connected Layer. Hasil pengujian dievaluasi dengan menggunakan data citra testing. Proses evaluasi dilakukan dengan menggunakan confusion matrix. Berdasarkan hasil pengujian aplikasi yang dirancang dengan akurasi model 99,4% dan akurasi validasi 97,8% maka aplikasi berguna untuk mendeteksi penyakit apel dengan menggunakan citra daun apel.

Kata kunci: penyakit daun apel; jaringan saraf convolutional; klasifikasi citra; LeNet-5

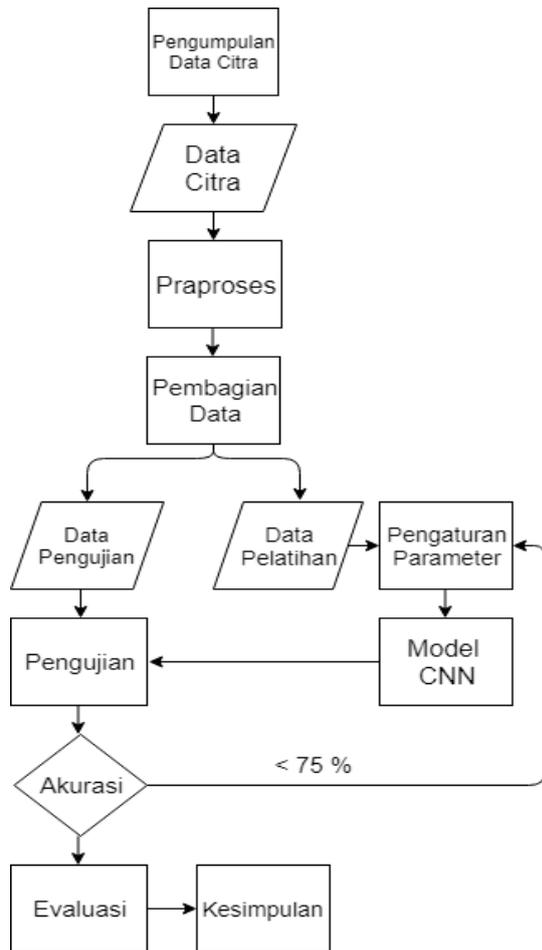
© 2020 Jurnal JOINTECS

1. Pendahuluan

Industri pertanian menjadi salah satu komoditas untuk mendorong pertumbuhan ekonomi di Indonesia. Salah satu tanaman yang menjadi komoditas perekonomian adalah tanaman apel. Menurut data statistik tanaman buah dan sayuran tahun 2017 yang dipublikasikan oleh

BPS, jumlah total produksi apel pada tahun 2017 sebesar 319004 ton yang merupakan salah satu produksi tanaman buah terbesar di Indonesia [1]. Terdapat banyak penyakit yang dapat menyerang tanaman apel, oleh karena itu pendeteksian dini serta pengidentifikasian [2] penyakit tanaman mejadi faktor

Diterima Redaksi : 30-12-2019 | Selesai Revisi : 04-01-2020 | Diterbitkan Online : 26-01-2020



Gambar 1. Diagram Sistem

utama untuk mencegah dan mengurangi penyebaran penyakit tanaman apel. Dari permasalahan yang ada, penelitian ini bertujuan untuk mendeteksi penyakit apel dengan menggunakan citra daun apel yang pada prosesnya data *training* dilatih dengan melakukan 3 kali epoch yaitu 50, 75 dan 100 untuk menghasilkan tingkat akurasi yang maksimal.

Artificial intelligence atau kecerdasan buatan menjadi topik yang banyak dibicarakan dalam beberapa tahun terakhir. Industri pertanian merupakan salah satu sektor yang mampu mengimplementasikan kecerdasan buatan. Penggunaan teknologi [3] ini dapat mengetahui penyakit apel dengan membaginya ke dalam kelas tertentu. Untuk melakukannya dibutuhkan metode dengan kemampuan mengolah citra, salah satu teknik yang dapat digunakan adalah teknik *deep learning* dengan metode *Convolutional Neural Network* (CNN). Metode ini dapat menghasilkan tingkat akurasi yang signifikan karena metode ini mampu mempelajari sendiri fitur yang terdapat pada citra yang kompleks[4]. Penelitian ini memiliki batasan diantaranya citra tidak dapat diolah secara realtime, sebagai gantinya data citra diunggah kemudian diproses untuk diprediksi.

Metode CNN berhasil diterapkan pada bidang pertanian untuk mengidentifikasi penyakit tanaman apel

berdasarkan daun dengan tingkat akurasi sebesar 98.54% [5]. Metode CNN juga mampu mengidentifikasi penyakit apel menggunakan daun dengan tingkat akurasi sebesar 97.62% [6]. Pada kasus lain, metode CNN dapat mengidentifikasi penyakit tanaman apel dengan tingkat akurasi sebesar 97.58% [7].

Tabel 1. Dataset Daun Apel

Labels	Apple Scab	Black Rot	Cedar Apple Rust	Healthy	Total
Training Set	500	493	216	1312	2521
Testing Set	125	123	54	328	630

Metode CNN digunakan pada penelitian ini dengan arsitektur LeNet-5 yang dapat mengolah 3151 data citra dengan batas tingkat akurasi minimum sebesar 75%.

2. Metode Penelitian

Pada penelitian ini, dilakukan berbagai tahapan yaitu pemilihan dataset, pembuatan arsitektur model, pengujian model, dan mengukur akurasi model seperti ditunjukkan pada Gambar 1.

2.1. Dataset

Penelitian ini menggunakan dataset yang berasal dari *PlantVillage* yang dibuat oleh *SP Mohanty CEO & Co-founder CrowdAI* dengan jumlah sebanyak 3151 citra daun yang telah diklasifikasi berdasarkan kelasnya masing-masing. Pada penelitian ini data yang diunggah ke aplikasi juga menggunakan spesifikasi yang sama dengan dataset. Dataset ini diambil dengan kamera beresolusi 8 MP yang memiliki ukuran citra 256 x 256 piksel format RGB dan memiliki ekstensi jpg. Dataset ini juga diambil dengan pencahayaan yang baik untuk menghindari noise pada citra daun.

Kelas terdiri dari satu kelas daun yang sehat dan tiga kelas daun yang terserang penyakit. Tabel 1 akan menjelaskan mengenai jenis penyakit yang akan diteliti beserta contoh citra daun apel yang telah dideteksi penyakitnya disertai dengan penyebab dan gejala yang timbul pada citra apel.

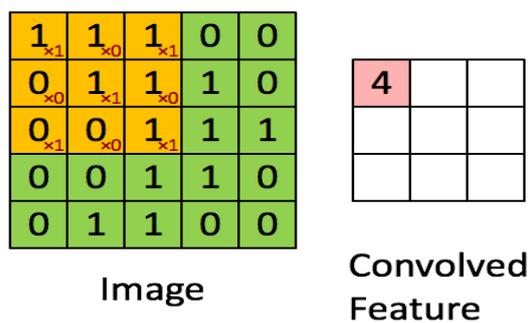
Dataset kemudian dibagi menjadi data training dan data testing dengan rasio 80:20 seperti ditunjukkan pada Tabel 2.

2.2. Convolutional Neural Network

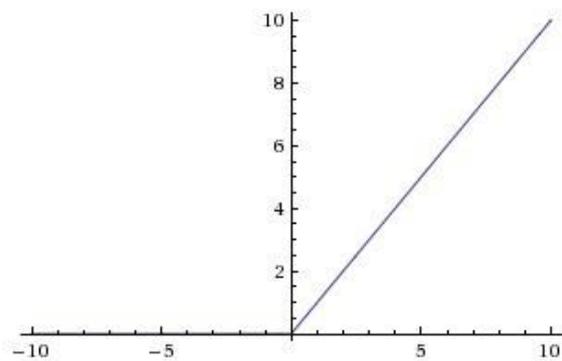
Penelitian ini menggunakan metode *Convolutional Neural Network* [8] dengan memakai arsitektur LeNet-5. Pada Gambar 2 merupakan arsitektur LeNet-5 yang dikembangkan oleh Yann LeCun. Arsitektur ini terdiri dari *Convolution Layer*, *subsampling* dan *fully connected layer*[9]. CNN terbagi menjadi beberapa layer sesuai dengan arsitektur LeNet-5.

Tabel 2. Tipe Penyakit

Penyakit	Citra	Penyebab	Gejala
<i>Apple Scab</i>		Jamur <i>Venturia inaequalis</i>	Bercak-bercak hijau di kedua sisi daun yang seiring berkembangnya penyakit, daun menjadi berwarna hitam keunguan
<i>Black Rot</i>		Jamur <i>Botryosphaeria obtusa</i>	Bercak-bercak ungu di permukaan daun dengan diameter 0,2 sampai 0,125 inci
<i>Cedar-Apple Rust</i>		Jamur <i>Gymnosporangium juniperi-virginianae</i>	Bercak-bercak coklat dan membuat daun menjadi rapuh



Gambar 2. Contoh Proses Konvolusi



Gambar 3. Rectified Linear Unit (ReLU)

2.3. Convolutional Layer

Konvolusi adalah proses operasi untuk menghasilkan *feature map* dengan menggunakan kernel secara berulang. Konvolusi memiliki kernel (kotak kuning) yang telah ditentukan sebelumnya dan mengaplikasikannya pada citra secara *offset*. Terdapat juga *feature map* atau *convolved feature* yang merupakan hasil dari proses konvolusi. Kernel akan bergerak dari kiri atas dan diulang terus sampai ke bagian paling kanan bawah. Proses ini juga menggunakan *stride* sebagai jarak pergerakan kernel, semakin kecil *stride* yang didefinisikan, semakin lama proses terjadinya konvolusi dan menghasilkan *feature map* yang lebih kecil[10].

Gambar 3 menunjukkan proses terjadinya konvolusi, image yang digambarkan dengan warna hijau akan dikonvolusi dengan menggunakan kernel yang digambarkan dengan warna kuning. Kernel akan bergerak dari kiri atas ke kanan bawah untuk menghasilkan *convolved feature*[11].

ReLU adalah fungsi aktivasi untuk memberikan kemampuan *network* agar dapat melakukan tugas-tugas yang non-linear[12]. Fungsi ReLU ditunjukkan rumus 1.

$$f(x) = \max(0, x) \quad (1)$$

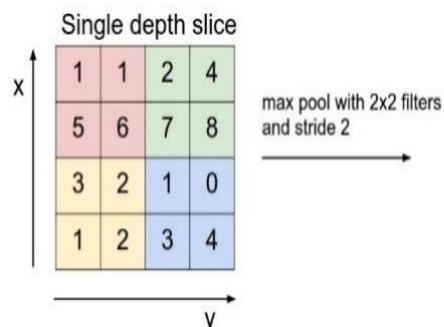
Jika Fungsi ReLU dijabarkan seperti rumus 2.

$$f(x) = \begin{cases} x, & x > 0 \\ 0, & x \leq 0 \end{cases} \quad (2)$$

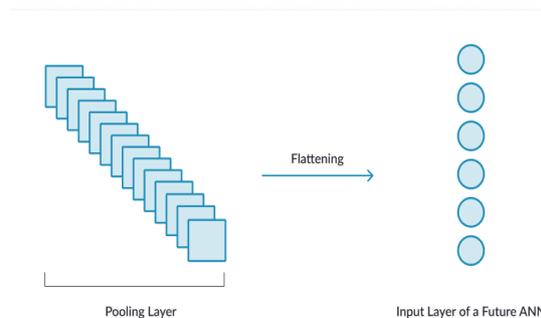
Keterangan :

- $f(x)$ = fungsi ReLU
- Jika nilai x lebih besar 0, maka nilai x akan tetap
- Jika nilai x lebih kecil atau sama dengan 0, maka nilai x akan dinaikkan menjadi 0

Pada rumus 1, $f(x)$ merupakan parameter dari fungsi ReLU yang membuat pembatas pada bilangan nol, yang berarti apabila $x \leq 0$ maka $x = 0$ dan jika $x > 0$ maka $x = x$ [13].



Gambar 4. Maxpooling



Gambar 5. Flattening

Pada Gambar 4 menunjukkan fungsi aktivasi ReLU, apabila nilai dibawah 0 atau negatif, maka angka akan dinaikkan menjadi 0, dan apabila nilai diatas angka 0 atau positif maka nilai akan tetap sesuai dengan yang ditunjukkan pada garis biru[14].

Subsampling merupakan proses untuk mengurangi ukuran *feature map* dari hasil konvolusi. Salah satu operasi *pooling* yang sering digunakan adalah *maxpooling*. *Subsampling* bertujuan mengurangi ukuran dari *feature map* agar proses komputasi menjadi lebih cepat[15].

Pada Gambar 5 merupakan contoh *maxpooling* dengan ukuran filter 2×2 . *Maxpooling* mengambil nilai terbesar dari *feature map* dan bertujuan untuk mengurangi ukuran dari *feature map*.

Flattening adalah teknik untuk merubah hasil dari proses yang memiliki ukuran dua dimensi menjadi satu dimensi vektor. *Flattening* bertujuan sebagai langkah awal inputan untuk masuk ke proses selanjutnya yaitu *neural network*[16].

Pada Gambar 6 menunjukkan proses flattening yang pada awalnya layer memiliki dua dimensi kemudian dilakukan proses flattening untuk menghasilkan satu dimensi vektor.

Fully Connected Layer mirip dengan cara bahwa neuron disusun dalam jaringan saraf tradisional. Tersusun dari input layer yang berasal dari proses *flattening*, *hidden layer*, serta *output layer*. Oleh karena itu, setiap node di lapisan sepenuhnya terhubung langsung ke setiap node baik sebelumnya dan di lapisan berikutnya[17].

Pada Gambar 7 menunjukkan proses dari *fully connected layer* yang terdiri dari input layer, hidden layer dan output layer.

2.4. Implementasi Model

Langkah selanjutnya adalah mengimplemetasikan model CNN sesuai dengan arsitektur LeNet-5. Penelitian ini menggunakan model arsitektur LeNet-5

dengan mengubah parameter agar meningkatkan hasil akurasi, yaitu menggunakan 32 maps yang berukuran 5×5 (C1) dengan *maxpooling* 2×2 (S1). Selanjutnya menggunakan 64 maps yang berukuran 5×5 (C2) dengan *maxpooling* 2×2 (S2). Kemudian dilanjutkan dengan memakai 2 *hidden layer*, *hidden layer* pertama memiliki 280 neuron dan pada *hidden layer* kedua memiliki 150 neuron.

Setelah model didapatkan, model akan dievaluasi untuk mengetahui tingkat keakurasian model dengan *confusion matrix recall* yang ditunjukkan rumus 3.

$$Recall = \frac{true\ positive}{true\ positive + false\ negative} \quad (3)$$

Keterangan :

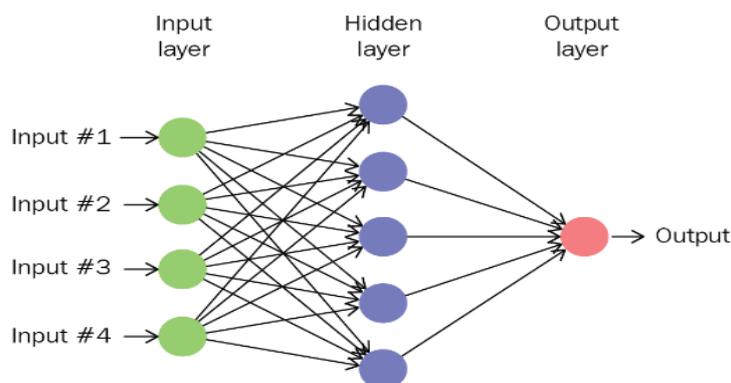
- *True positive* = output kelas positif yang berhasil ditebak sebagai kelas positif
- *False negative* = output kelas positif yang salah ditebak sebagai kelas negatif.

Pada rumus 3, Recall memiliki dua parameter untuk perhitungannya yaitu *true positive* yang merupakan output kelas positif yang ditebak sebagai kelas positif dan *true negative* yang merupakan output kelas positif yang ditebak sebagai kelas negatif. True positif akan dibagi dengan semua kelas output (*true positive* dan *true negative*) untuk mendapatkan hasil evaluasi pada model yang telah dibuat[18].

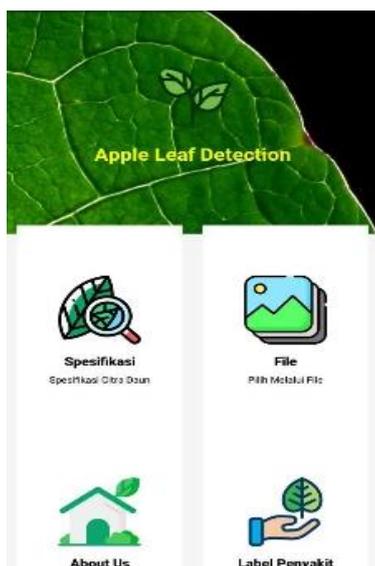
2.5. Implementasi Aplikasi

Pada penelitian ini, aplikasi dibuat dengan menggunakan flask sebagai *backend* dan *react-native* sebagai *frontend* dengan *Application Programming Interface* (API) yang digunakan sebagai server dari model *deep learning*. Tahapan yang dilakukan adalah memilih dan mengunggah citra daun apel untuk menuju server *deep learning*.

Aplikasi terdiri dari halaman tampilan utama dan juga halaman untuk memprediksi penyakit pada daun tanaman apel. Aplikasi ini memprediksi nama penyakit, gejala penyakit dan penanganan pada tanaman apel.



Gambar 6. Fully Connected Layer



Gambar 7. Halaman Utama Aplikasi



Gambar 8. Halaman Prediksi

Pada Gambar 8 merupakan tampilan user interface untuk aplikasi pendeteksi penyakit pada daun tanaman apel yang terdiri dari spesifikasi citra, file, about us, dan label penyakit Pada Gambar 9 merupakan halaman prediksi yang terdiri dari nama penyakit daun apel, gejala dan penanganan.

Pada proses penginputan citra, citra yang dapat diterima oleh server adalah file yang memiliki ekstensi jpg. Ukuran file yang direkomendasikan adalah 256 x 256 piksel dengan warna RGB, diambil dengan kamera beresolusi minimum 8 MP dengan pencahayaan yang baik dan memiliki fokus terhadap citra daun apel.

3. Hasil dan Pembahasan

Penelitian ini memiliki hasil tingkat akurasi yang akan diuji menggunakan perangkat lunak dan perangkat keras yang akan ditunjukkan pada Tabel 3 dan Tabel 4. Pada penelitian ini, dilakukan 3 kali percobaan iterasi epoch untuk mencari akurasi terbaik yang ditampilkan pada Tabel 5. Hasil dari Tabel 5 dapat dilihat bahwa hasil yang didapatkan untuk epoch 50, 75 dan 100 memiliki akurasi yang beragam dengan loss model yang

menurun seiring dengan bertambahnya jumlah epoch, ini menandakan bahwa arsitektur yang telah digunakan cukup baik untuk dilakukan ke proses selanjutnya yaitu pengujian data atau validasi. Gambar 10 merupakan grafik untuk memvisualisasikan akurasi model dan validasi pada epoch 100.

Pada Gambar 10 menunjukkan grafik dari akurasi model yang terdiri dari model training dan model testing atau validasi untuk 100 epoch. Pada model training ditunjukkan oleh garis grafik dengan warna biru yang memiliki akurasi sebesar 99.4% dan pada model testing atau validasi ditunjukkan oleh garis grafik dengan warna merah yang memiliki akurasi sebesar 97.8%. Hasil pengujian selanjutnya akan dievaluasi dengan menggunakan data citra berjumlah

Dari Tabel 6, menunjukkan tingkat akurasi tertinggi adalah kelas Black Rot dengan tingkat akurasi 100% dan akurasi terendah berada di kelas Apple Scab dengan tingkat akurasi sebesar 74,4%. Sehingga didapatkan tingkat akurasi rata-rata untuk hasil evaluasi sebesar 89,62%.

Tabel 3. Perangkat Lunak

Perangkat Lunak	Keterangan	Versi
Sistem Operasi	Windows	10
Text Editor	Spyder	3.3.6
	Visual Studio Code	1.40.2
Bahasa Pemrograman	Python	3.7.4
	Javascript	1.8.5
	React Native	0.61
Library / Framework	Tensorflow	1.14.0
	Keras	2.2.4
	Flask	1.1.1
	Matplotlib	3.1.1
	Numpy	1.17.3

Tabel 4. Perangkat Keras

Perangkat Keras	Keterangan
Processor	Intel i5-4200U 1.60GHZ
RAM	12 GB
Graphics	NVIDIA GeForce GT 740M



Gambar 9. Hasil Akurasi Model Pada 100 Epoch

Tabel 5. Hasil Klasifikasi

Epoch	Akurasi Model	Loss Model	Akurasi Validasi	Loss Validasi	Waktu
50	99%	0,09	91,2%	0,32	6 jam 15 menit
75	99,3%	0,06	95,6%	0,28	9 jam 30 menit
100	99,4%	0,04	97,8%	0,23	12 jam 10 menit

Tabel 6. Hasil Evaluasi dengan Confusion Matrix

Aktual	Prediksi				Akurasi
	Apple Scab	Black Rot	Cedar Apple Rust	Healthy	
Apple Scab	158	28	9	5	79%
Black Rot	0	195	5	0	97.5%
Cedar Apple Rust	0	2	197	1	98.5%
Healthy	1	32	0	167	83.5%
	Total				89.62%

4. Kesimpulan

Pada data training untuk epoch 50, 75 dan 100 menghasilkan rata-rata akurasi model sebesar 99,2% dan memiliki rata-rata loss model sebesar 0,063. Dari data testing didapat hasil untuk epoch 50, 75 dan 100 menghasilkan rata-rata akurasi validasi sebesar 94,9% dan memiliki rata-rata loss validasi sebesar 0,277. Berdasarkan aplikasi yang dirancang dengan akurasi model 99,4% dan akurasi validasi 97,8% maka aplikasi ini berguna untuk mendeteksi penyakit apel dengan menggunakan citra daun apel.

Daftar Pustaka

- [1] Badan Pusat Statistik, "Statistik Tanaman Buah-buahan dan Sayuran Tahunan Indonesia 2017," BPS, 2018.
- [2] I. Imanuddin, F. Alhadi, R. Oktafian, and A. Ihsan, "Deteksi Mata Mengantuk pada Pengemudi Mobil Menggunakan Metode Viola Jones," *MATRIK J. Manajemen, Tek. Inform. dan Rekayasa Komput.*, vol. 18, no. 2, pp. 321–329, 2019.
- [3] A. Nidomudin, A. P. Nugroho, and M. N. Cholis, "Sistem Pakar Deteksi Tingkat Kesuburan Tanah Menggunakan Fuzzy Logic," *JOINTECS (Journal Inf. Technol. Comput. Sci.)*, vol. 2, no. 2, pp. 79–

- 84, 2017.
- [4] M. G. Arizqia and A. A. Widodo, "Rancang Bangun Aplikasi Dengan Linear Congruent Method (LCM) Sebagai Pengacakan Soal," *JOINTECS (Journal Inf. Technol. Comput. Sci.*, vol. 2, no. 1, pp. 1–6, 2017.
- [5] S. Baranwal, S. Khandelwal, and A. Arora, "Deep Learning Convolutional Neural Network for Apple Leaves Disease Detection," *SUSCOM*, pp. 260–267, 2019.
- [6] B. Liu, "Identification of Apple Leaf Diseases Based on Deep Convolutional Neural Networks," *Symmetry (Basel)*, 2018.
- [7] T. Fang, P. Chen, J. Zhang, and B. Wang, "Identification of Apple Leaf Diseases Based on Convolutional Neural Network," *Springer*, vol. 2, pp. 553–564, 2019.
- [8] M. Agarwal, "FCNN-LDA : A Faster Convolution Neural Network model for Leaf Disease identification on Apple's Leaf Dataset," *2019 12th Int. Conf. Inf. Commun. Technol. Syst.*, pp. 246–251, 2019.
- [9] J. Liu, S. Yang, Y. Cheng, and Z. Song, "Plant Leaf Classification Based on Deep Learning," *2018 Chinese Autom. Congr.*, pp. 3165–3169, 2018.
- [10] A. Mg, J. Hanson, A. Joy, and J. Francis, "Plant Leaf Disease Detection using Deep Learning and Convolutional Neural Network," *Int. J. Eng. Sci. Comput.*, vol. 7, no. 3, 2017.
- [11] V. Suma, R. A. Shetty, R. F. Tated, S. Rohan, and T. S. Pujar, "CNN based Leaf Disease Identification and Remedy Recommendation System," *2019 3rd Int. Conf. Electron. Commun. Aerosp. Technol.*, pp. 395–399, 2019.
- [12] P. Jiang, Y. Chen, and B. I. N. Liu, "Real-Time Detection of Apple Leaf Diseases Using Deep Learning Approach Based on Improved Convolutional Neural Networks," *IEEE Access*, vol. 7, pp. 59069–59080, 2019.
- [13] H. Ide, "Improvement of Learning for CNN with ReLU Activation by Sparse Regularization," *IEEE Access*, pp. 2684–2691, 2017.
- [14] Y. Lu, S. Yi, N. Zeng, Y. Liu, and Y. Zhang, "Identification of Rice Diseases Using Deep Convolutional Neural Networks," *Neurocomputing*, vol. 267, pp. 378–384, 2017.
- [15] L. G. Nachtigall and R. M. Araujo, "Classification of Apple Tree Disorders Using Convolutional Neural Networks," *IEEE Access*, pp. 472–476, 2016.
- [16] K. P. Ferentinos, "Deep learning models for plant disease detection and diagnosis," *Comput. Electron. Agric.*, vol. 145, no. January, pp. 311–318, 2018.
- [17] M. Francis, "Disease Detection and Classification in Agricultural Plants Using Convolutional Neural Networks – A Visual Understanding," *2019 6th Int. Conf. Signal Process. Integr. Networks*, pp. 1063–1068, 2019.
- [18] K. Park and J. Lee, "Classification of apple leaf conditions in hyper-spectral images for diagnosis of Marssonina blotch using mRMR and deep neural network," *Comput. Electron. Agric.*, vol. 148, no. July 2017, pp. 179–187, 2018.

Halaman ini sengaja dikosongkan