

Identifikasi Penyakit pada Citra Daun Kentang Menggunakan Convolutional Neural Network (CNN)

Alang Mulya Lesmana¹⁾, Ronna Putri Fadhillah²⁾, Chaerur Rozikin³⁾

¹⁾²⁾³⁾ Prodi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Singaperbangsa Karawang
Jl. H.S. Ronggowaluyo Kel.Puseurjaya, Telukjambe Timur, Kab.Karawang, Jawa Barat

¹⁾ alang.mulya18074@student.unsika.ac.id

²⁾ ronna.fadhillah18111@student.unsika.ac.id

²⁾ chaerur.rozikin@staff.unsika.ac.id

Abstrak

Kentang merupakan produk hasil pertanian yang memiliki kandungan tepung tertinggi keempat setelah jagung, gandum, dan padi. Pengelolaan pertanian kentang memiliki beberapa permasalahan diantaranya adalah penyakit pada daun kentang yang apabila tidak diselesaikan dapat menyebabkan hasil produksi yang buruk hingga gagal panen. Penyakit yang sering ditemui pada daun kentang yaitu *early blight* dan *late blight*. Kedua penyakit ini memiliki gejala serta penanganan yang berbeda, namun proses identifikasi yang lambat dapat menyebabkan tambahan biaya untuk perawatan. Pada penelitian ini penulis memanfaatkan algoritma deep learning yaitu Convolutional Neural Network (CNN) untuk identifikasi citra pada daun kentang. Metode Convolutional Neural Network (CNN) memanfaatkan proses konvolusi dimana citra akan dipecah menjadi gambar yang lebih kecil dengan konvolusi yang sama. Hasil dari masing-masing gambar yang lebih kecil tersebut kemudian dimasukkan kedalam array baru yang nantinya akan digunakan untuk prediksi. Data yang digunakan pada penelitian ini berjumlah 5400 citra yang terbagi menjadi 3 kelas yaitu citra sehat, citra *early blight*, dan citra *late blight*. Hasil pengujian menunjukkan akurasi tertinggi pada *data validation* yaitu sebesar 99% sehingga dapat disimpulkan bahwa algoritma *deep learning* Convolutional Neural Network (CNN) dapat melakukan proses identifikasi penyakit pada citra daun kentang dengan baik.

Kata kunci: identifikasi citra, daun kentang, *early blight*, *late blight*, Convolutional Neural Network (CNN)

Abstract

Potatoes are agricultural products that have the fourth highest flour content after corn, wheat, and rice. The management of potato farming has several problems including disease on potato leaves which if not resolved can lead to poor production yields to crop failure. Diseases that are often found in potato leaves are an early blight and late blight. These two diseases have different symptoms and treatments, but the slow identification process can lead to additional costs for treatment. In this study, the author uses a deep learning algorithm, namely a convolutional neural network (CNN) to identify images on potato leaves. The convolutional neural network (CNN) method utilizes a convolution process where the image will be split into smaller images with the same convolution. The results of each of the smaller images are then entered into a new array which will be used for prediction. The data used in this study amounted to 5400 images which were divided into 3 classes, namely healthy images, early blight images, and late blight images. The test results show the highest accuracy in data validation, which is 99%, so it can be concluded that the deep learning convolutional neural network (CNN) algorithm can carry out the disease identification process in potato leaf images well.

Keywords: image identification, potato leaf, *early blight*, *late blight*, Convolutional Neural Network (CNN)

1. PENDAHULUAN

Kentang merupakan umbi-umbian yang memiliki kandungan karbohidrat. Selain itu kentang juga merupakan salah satu tanaman pangan yang tumbuh subur di dataran tinggi Indonesia, serat dan vitamin C yang baik bagi tubuh terkandung di dalamnya [1]. Dalam membudidayakan suatu tanaman harus memperhatikan beberapa hal demi kelangsungan produksi tanaman berjalan dengan baik, termasuk dalam membudidayakan tanaman kentang. Tetapi pada saat membudidayakan tentu tidak terlepas dari permasalahan tanaman kentang yang terserang penyakit. Petani yang tidak cermat dalam mencermati serangan penyakit pada daun tanaman kentang, penyakit adalah salah satu aspek utama menurunnya kualitas serta kuantitas penciptaan pertanian secara global [2].

Petani sayuran adalah ujung tombak dalam menghasilkan yang baik bagi hasil pertanian sayuran, terutama petani kentang [3]. Petani perlu meningkatkan pengetahuan dalam mencegah, mengidentifikasi, dan menangani tanaman kentang yang belum atau yang sudah terserang penyakit. Penyakit yang paling umum ditemukan pada daun tanaman kentang yaitu busuk daun (*late blight*) dan bercak kering (*early blight*). Salah satu faktor penyebab terjadinya penyakit busuk daun berkembang yaitu tempat dingin dan lembab [4]. Dengan melakukan identifikasi kedua penyakit pada daun tanaman kentang ini diharapkan para petani dapat mencegah terjadinya serangan penyakit pada tanaman kentang, sehingga tidak akan terjadi lagi keterhambatan produksi dan pengurangan hasil produksi.

Terdapat pada penelitian sebelumnya yang membahas tentang penerapan algoritma *Convolutional Neural Network* terhadap diagnosis penyakit sawit yang menguji sebanyak 2490 citra menghasilkan nilai akurasi tertinggi sebesar 89% [5]. Penelitian lain dilakukan dengan sedikit modifikasi yaitu dengan penerapan *Convolutional Neural Network* dengan arsitektur MobileNet pada identifikasi penyakit pada tanaman gambas menghasilkan nilai akurasi sebesar 90% [6]. Penelitian lain pun dilakukan dengan menggunakan algoritma *Convolutional Neural Network* untuk mengklasifikasi 6 penyakit daun dan 2 daun sehat pada daun apel dan anggur yang menghasilkan nilai akurasi dengan rata-rata 79,25% dan nilai akurasi tertinggi yaitu sebesar 94,767% [7]. Selain itu algoritma CNN pun pada penelitian sebelumnya digunakan untuk mendeteksi Covid-19 berdasarkan citra X-Ray yang dibuat berdasarkan model VGG16 melalui *transfer learning* didapati hasil akurasi yang sangat baik yaitu sebesar 100% dengan menggunakan 24 data [8]. Penelitian yang lain pun dalam mengklasifikasi jerawat berdasarkan jenisnya menggunakan CNN dan menggunakan data *train* 90% dan data *test* 10%, dua lapisan konvolusi, lapisan konvolusi pertama 10 filter dan lapisan konvolusi kedua 20 filter, dan menghasilkan nilai akurasi sebesar 91,6% [9]. Dari beberapa penelitian tersebut bisa dapat disimpulkan bahwa penggunaan algoritma *Convolutional Neural Network* dalam identifikasi penyakit pada tanaman cukup baik.

Dilihat dari permasalahan tersebut untuk menekan angka kegalalan dalam budidaya kentang teknologi perangkat lunak untuk membantu mendeteksi penyakit lebih dini pada kentang untuk membantu petani dalam budidaya tanaman kentang sangat dibutuhkan [10]. Maka dari itu seiring berkembangnya teknologi, penulis menggunakan bahasa pemrograman python untuk mengidentifikasi penyakit pada daun tanaman kentang dengan menggunakan algoritma *Convolutional Neural Network* (CNN). Tujuan dari penelitian ini yaitu identifikasi penyakit daun kentang ke dalam kelas sehat, busuk daun (*late blight*) atau bercak kering (*early blight*).

2. TINJAUAN PUSTAKA

2.1 Penyakit Tanaman Kentang

Kentang (*Solanum Tuberosum*) merupakan suatu tanaman sayuran yang menghasilkan umbi dari famili Solanacea [11], sejenis tumbuhan herba semusim (*annual crop*) yang tingginya dapat mencapai 100 cm. Umbi kentang dapat dijadikan bahan pangan karena memiliki kandungan karbohidrat, mineral, kalori serta vitamin yang cukup besar yang bisa menggeser posisi bahan pangan karbohidrat yang bersumber dari beras, gandum maupun jagung guna memenuhi kebutuhan pangan yang telah populer di dunia [12]. Kentang berdasarkan jenisnya dibagi menjadi 2 kelompok ialah yang menyesuaikan diri pada penyinaran pendek (*short day plant*) yang

biasanya ditanam di wilayah Andes, dinamakan Andigena, dan jenis *Tuberosum* (tipe Chili) yang membutuhkan penyinaran panjang (*long day plant*), jenis ini merupakan jenis utama yang ditanam diseluruh dunia [13].

Penyakit pada tanaman kentang beberapa disebabkan oleh bakteri, virus, maupun cendawan [14]. Namun, penyakit tanaman kentang umum terjadi akibat beberapa patogen yang menyerang tanaman kentang, faktor seperti iklim dan cuaca pun bisa menjadi salah satu penyebab munculnya penyakit pada tanaman kentang. Jika tidak dicegah dan ditangani dengan baik maka bisa akan berpengaruh pada penurunan hasil produksi.

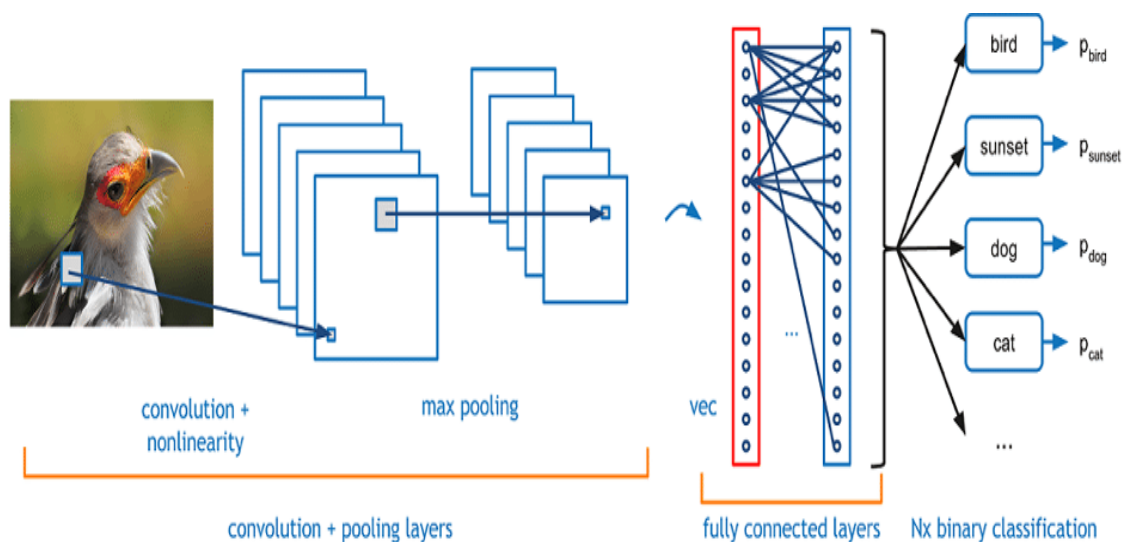
Penyakit daun busuk (*late blight*) disebabkan oleh patogen *Phytophthora infestans* (Mont.) de Bary; Ordo Pythiales; Family Pythiaceae. Penyakit ini menyerang pada daun tanaman kentang [15]. Ciri-ciri daun yang terkena penyakit ini diantaranya: daun memiliki bercak nekrotis pada tepi dan ujungnya, dan jika pada suhu yang tidak terlalu rendah dan kelembaban cukup tinggi bercak-bercak tersebut akan meluas dengan cepat dan akan mematikan seluruh daun.

Penyakit bercak kering (*early blight*) disebabkan oleh jamur *alternaria solani*, jamur ini menyerang daun, batang, serta umbi-umbian. Fase daun merupakan yang sangat umum dan mengganggu. Kehilangan hasil kentang sampai 50% yang diakibatkan oleh kerusakan EB sudah terjadi di seluruh dunia [16]. Ciri-ciri daun yang terkena penyakit ini diantaranya, munculnya bercak abu-abu hingga coklat di daun dan perlahan-lahan tumbuh secara konsentris di sekitar bagian tengah yang bersih dan berbentuk mata banteng.

2.2 Convolutional Neural Network

Convolutional Neural Network merupakan salah satu jenis algoritma *supervised* yang cara kerjanya adalah menerima *input* berupa gambar [17]. CNN sangat cocok untuk pemrosesan dengan input berupa citra. karena pada CNN biasa digunakan untuk mengklasifikasikan citra [18]. CNN adalah suatu metode *machine learning* yang dikembangkan dari *Multi Layer Perceptron* (MLP) yang didesain untuk mengolah data 2 dimensi [19].

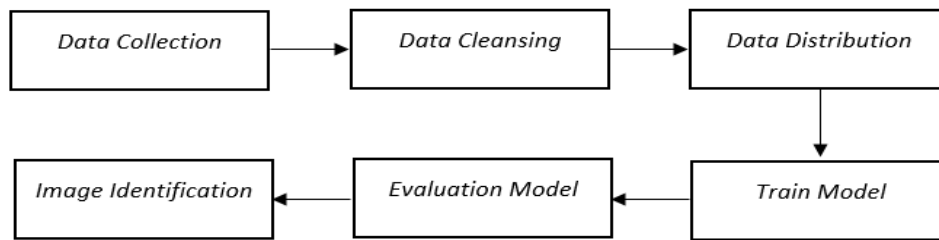
Cara kerja dari algoritma *Convolutional Neural Network* ditunjukkan pada Gambar 1.



Gambar 1. Cara kerja *convolutional neural network*

3. METODE PENELITIAN

Suatu penelitian membutuhkan langkah-langkah penelitian yang tersusun dengan baik sehingga proses penelitian dapat mencapai tujuan dengan baik. Adapun langkah-langkah penelitian yang dilakukan tercantum pada Gambar 2.



Gambar 2. Metode penelitian

3.1 Data Collection

Data yang digunakan merupakan data citra yang dapat diakses secara terbuka di situs *kaggle* dengan total jumlah data sebanyak 5702 citra yang terbagi menjadi 3 kategori, yakni citra sehat, citra penyakit *early blight*, dan citra penyakit *late blight*. Jumlah data pada masing-masing kategori nya adalah 1824 citra sehat, 1939 citra *early blight*, dan 1939 citra *late blight*.

3.2 Data Cleansing

Data Cleansing mengacu pada mengidentifikasi dan memperbaiki kesalahan dalam kumpulan data yang dapat berdampak negatif pada model prediktif [20]. Persebaran data yang tidak merata pada setiap kategori serta banyaknya citra yang kurang jelas dapat memicu terjadinya kesalahan dalam proses klasifikasi citra. Permasalahan tersebut diatasi dengan cara menyeleksi data yang kurang jelas dan menyamakan persebaran data untuk setiap kategori nya. Jumlah persebaran data setelah proses ini adalah 1800 data untuk masing-masing kategori.

3.3 Data Distribution

Pembagian data *training* dan data *validation* diperlukan untuk melatih model yang akan digunakan untuk klasifikasi citra. Data akan dibagi menjadi dua, yaitu data *training* yang berfungsi untuk melatih model dan data *validation* yang berfungsi untuk memvalidasi model yang telah diuji. Pembagian data dilakukan sebanyak tiga kali dengan skenario yang berbeda. Pada skenario 1, data dibagi menjadi 90:10 untuk data training dan data validasi, pada scenario kedua data dibagi menjadi 80:20 untuk data training dan data validasi, dan pada scenario ketiga, data dibagi menjadi 70:30 untuk data training dan data validasi. Adapun pembagian data dan jumlah citra dari masing-masing skenario dapat dilihat pada Tabel 1.

Tabel 1. Distribusi data

Skenario	Perbandingan Data	Data Training	Data Validation
1	90:10	4860	540
2	80:20	4320	1080
3	70:30	3780	1620

3.4 Train Model

Penelitian ini dilakukan menggunakan model *deep learning* dengan arsitektur CNN untuk proses identifikasi penyakit pada citra daun kentang. Model dilatih dengan 25 *steps* per *epoch* dan 4 *validation steps*, setiap citra akan diubah menjadi 150x150 piksel. Model yang digunakan pada penelitian dapat dilihat pada Tabel 2.

Tabel 2. Model *Convolutional Neural Network* (CNN)

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 148, 148, 16)	448
max_pooling2d (MaxPooling2D)	(None, 74, 74, 16)	0
conv2d_1 (Conv2D)	(None, 72, 72, 32)	4640
max_pooling2d_1 (MaxPooling2D)	(None, 36, 36, 32)	0
conv2d_2 (Conv2D)	(None, 34, 34, 64)	18496

max_pooling2d_1 (MaxPooling2D)	(None, 17, 17, 64)	0
flatten (Flatten)	(None, 18496)	0
dense (Dense)	(None, 512)	9470464
dense_1 (Dense)	(None, 3)	1539

Total params : 9,495,587

Trainable params : 9,495,587

Non-trainable params : 0

4. PEMBAHASAN

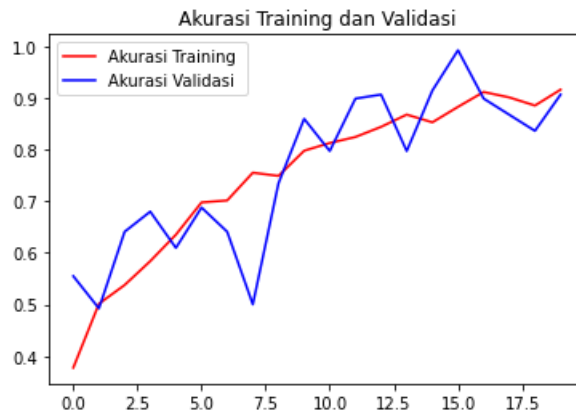
Model diuji dengan 3 skenario berbeda untuk melihat hasil terbaik yang bisa didapatkan, masing-masing skenario akan diuji sebanyak 20 *epoch* dengan 25 *steps* per *epoch* dan 4 *validation steps* dengan *train model* yang telah dibuat sebelumnya. Citra untuk data training dan data validasi akan diubah menjadi 150x150 piksel untuk mempercepat proses uji coba. Hasil dari ujicoba masing-masing skenario adalah sebagai berikut.

4.1 Hasil Uji Coba Skenario 1

Pada skenario 1, citra dibagi menjadi dua dengan perbandingan 90:10 untuk data training dan data validasi. Jumlah citra yang digunakan adalah 4860 untuk data training, dan 540 untuk data validasi. Citra akan di *rescale* sebesar 1/255 dari nilai aslinya, rentangan ini berfungsi untuk mempermudah proses belajar dari jaringan syaraf tiruan. Selanjutnya scenario lain adalah ketika citra dirotasi sebesar 20 derajat, dilebarkan sebesar 20% dari ukuran aslinya, ditinggikan sebesar 20% dari ukuran aslinya, dipangkas sebesar 20% dari ukuran aslinya, diperbesar sebanyak 20%, dan jika citra di *flip* secara horizontal. Nilai piksel yang kosong pada citra akan diisi dengan nilai piksel terdekat. Hasil dari ujicoba pada skenario 1 menunjukkan akurasi tertinggi yaitu 91% untuk data *training* dan 99% untuk data *validation*. Proses uji coba skenario 1 dapat dilihat pada Gambar 3. dan grafik akurasi dapat dilihat pada Gambar 4.

```
Epoch 1/20  
25/25 - 251s - loss: 2.1346 - accuracy: 0.3775 - val_loss: 0.9805 - val_accuracy: 0.5547 - 251s/epoch - 10s/step  
Epoch 2/20  
25/25 - 205s - loss: 0.9550 - accuracy: 0.5013 - val_loss: 1.0117 - val_accuracy: 0.4922 - 205s/epoch - 8s/step  
Epoch 3/20  
25/25 - 156s - loss: 0.9516 - accuracy: 0.5375 - val_loss: 0.8170 - val_accuracy: 0.6406 - 156s/epoch - 6s/step  
Epoch 4/20  
25/25 - 130s - loss: 0.8842 - accuracy: 0.5838 - val_loss: 0.7080 - val_accuracy: 0.6797 - 130s/epoch - 5s/step  
Epoch 5/20  
25/25 - 116s - loss: 0.8136 - accuracy: 0.6350 - val_loss: 0.7145 - val_accuracy: 0.6094 - 116s/epoch - 5s/step  
Epoch 6/20  
25/25 - 100s - loss: 0.6911 - accuracy: 0.6975 - val_loss: 0.6257 - val_accuracy: 0.6875 - 100s/epoch - 4s/step  
Epoch 7/20  
25/25 - 78s - loss: 0.6282 - accuracy: 0.7013 - val_loss: 0.7040 - val_accuracy: 0.6406 - 78s/epoch - 3s/step  
Epoch 8/20  
25/25 - 72s - loss: 0.6036 - accuracy: 0.7550 - val_loss: 2.0261 - val_accuracy: 0.5000 - 72s/epoch - 3s/step  
Epoch 9/20  
25/25 - 62s - loss: 0.6241 - accuracy: 0.7487 - val_loss: 0.5426 - val_accuracy: 0.7344 - 62s/epoch - 2s/step  
Epoch 10/20  
25/25 - 51s - loss: 0.5103 - accuracy: 0.7975 - val_loss: 0.4292 - val_accuracy: 0.8594 - 51s/epoch - 2s/step  
Epoch 11/20  
25/25 - 43s - loss: 0.4856 - accuracy: 0.8125 - val_loss: 0.5176 - val_accuracy: 0.7969 - 43s/epoch - 2s/step  
Epoch 12/20  
25/25 - 42s - loss: 0.4821 - accuracy: 0.8241 - val_loss: 0.2896 - val_accuracy: 0.8984 - 42s/epoch - 2s/step  
Epoch 13/20  
25/25 - 40s - loss: 0.4082 - accuracy: 0.8438 - val_loss: 0.3202 - val_accuracy: 0.9062 - 40s/epoch - 2s/step  
Epoch 14/20  
25/25 - 35s - loss: 0.3807 - accuracy: 0.8675 - val_loss: 0.4743 - val_accuracy: 0.7969 - 35s/epoch - 1s/step  
Epoch 15/20  
25/25 - 28s - loss: 0.3691 - accuracy: 0.8525 - val_loss: 0.2717 - val_accuracy: 0.9141 - 28s/epoch - 1s/step  
Epoch 16/20  
25/25 - 29s - loss: 0.5065 - accuracy: 0.8825 - val_loss: 0.1201 - val_accuracy: 0.9922 - 29s/epoch - 1s/step  
Epoch 17/20  
25/25 - 29s - loss: 0.2581 - accuracy: 0.9112 - val_loss: 0.2392 - val_accuracy: 0.8984 - 29s/epoch - 1s/step  
Epoch 18/20  
25/25 - 28s - loss: 0.2695 - accuracy: 0.9008 - val_loss: 0.2806 - val_accuracy: 0.8672 - 28s/epoch - 1s/step  
Epoch 19/20  
25/25 - 26s - loss: 0.2985 - accuracy: 0.8850 - val_loss: 0.3449 - val_accuracy: 0.8359 - 26s/epoch - 1s/step  
Epoch 20/20  
25/25 - 27s - loss: 0.2275 - accuracy: 0.9158 - val_loss: 0.2232 - val_accuracy: 0.9062 - 27s/epoch - 1s/step
```

Gambar 3. Hasil uji coba skenario 1



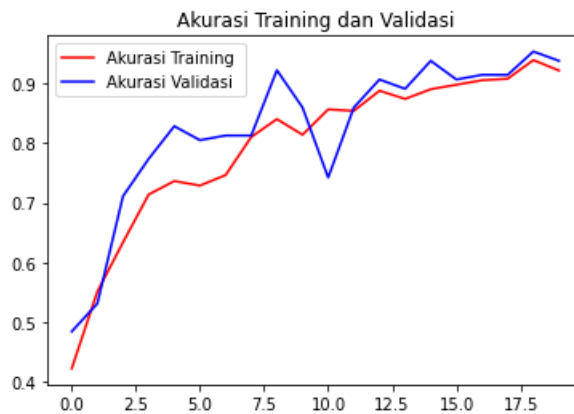
Gambar 4. Grafik hasil uji coba skenario 1

4.2 Hasil Uji Coba Skenario 2

Pada skenario 2, citra dibagi menjadi dua dengan perbandingan 80:20 untuk data training dan data validasi. Jumlah citra yang digunakan adalah 4320 untuk data training, dan 1080 untuk data validasi. Citra akan di *rescale* sebesar $1/255$ dari nilai aslinya, rentangan ini berfungsi untuk mempermudah proses belajar dari jaringan syaraf tiruan. Selanjutnya skenario lain adalah ketika citra dirotasi sebesar 20 derajat, dilebarkan sebesar 20% dari ukuran aslinya, ditinggikan sebesar 20% dari ukuran aslinya, dipangkas sebesar 20% dari ukuran aslinya, diperbesar sebanyak 20%, dan jika citra di *flip* secara horizontal. Nilai piksel yang kosong pada citra akan diisi dengan nilai piksel terdekat. Uji coba pada skenario 2 menunjukkan hasil akurasi tertinggi yaitu 93% untuk data *training* dan 95% untuk data *validation*. Proses uji coba dapat dilihat pada Gambar 5 dan grafik akurasi dapat dilihat pada Gambar 6.

```
Epoch 1/20  
25/25 - 28s - loss: 1.3752 - accuracy: 0.4225 - val_loss: 0.9815 - val_accuracy: 0.4844 - 28s/epoch - 1s/step  
Epoch 2/20  
25/25 - 26s - loss: 0.9011 - accuracy: 0.5512 - val_loss: 0.9012 - val_accuracy: 0.5312 - 26s/epoch - 1s/step  
Epoch 3/20  
25/25 - 25s - loss: 0.8275 - accuracy: 0.6338 - val_loss: 0.7421 - val_accuracy: 0.7109 - 25s/epoch - 1s/step  
Epoch 4/20  
25/25 - 26s - loss: 0.6788 - accuracy: 0.7138 - val_loss: 0.5342 - val_accuracy: 0.7734 - 26s/epoch - 1s/step  
Epoch 5/20  
25/25 - 26s - loss: 0.6464 - accuracy: 0.7362 - val_loss: 0.4965 - val_accuracy: 0.8281 - 26s/epoch - 1s/step  
Epoch 6/20  
25/25 - 25s - loss: 0.6397 - accuracy: 0.7287 - val_loss: 0.5329 - val_accuracy: 0.8047 - 25s/epoch - 999ms/step  
Epoch 7/20  
25/25 - 25s - loss: 0.5729 - accuracy: 0.7462 - val_loss: 0.4891 - val_accuracy: 0.8125 - 25s/epoch - 1s/step  
Epoch 8/20  
25/25 - 25s - loss: 0.4889 - accuracy: 0.8100 - val_loss: 0.5754 - val_accuracy: 0.8125 - 25s/epoch - 1s/step  
Epoch 9/20  
25/25 - 25s - loss: 0.4247 - accuracy: 0.8400 - val_loss: 0.3022 - val_accuracy: 0.9219 - 25s/epoch - 992ms/step  
Epoch 10/20  
25/25 - 25s - loss: 0.4784 - accuracy: 0.8138 - val_loss: 0.3582 - val_accuracy: 0.8594 - 25s/epoch - 996ms/step  
Epoch 11/20  
25/25 - 25s - loss: 0.3408 - accuracy: 0.8562 - val_loss: 0.5688 - val_accuracy: 0.7422 - 25s/epoch - 982ms/step  
Epoch 12/20  
25/25 - 24s - loss: 0.3819 - accuracy: 0.8537 - val_loss: 0.3796 - val_accuracy: 0.8594 - 24s/epoch - 973ms/step  
Epoch 13/20  
25/25 - 25s - loss: 0.3158 - accuracy: 0.8875 - val_loss: 0.2368 - val_accuracy: 0.9062 - 25s/epoch - 982ms/step  
Epoch 14/20  
25/25 - 24s - loss: 0.3827 - accuracy: 0.8737 - val_loss: 0.2655 - val_accuracy: 0.8906 - 24s/epoch - 974ms/step  
Epoch 15/20  
25/25 - 24s - loss: 0.2709 - accuracy: 0.8900 - val_loss: 0.1447 - val_accuracy: 0.9375 - 24s/epoch - 970ms/step  
Epoch 16/20  
25/25 - 24s - loss: 0.3492 - accuracy: 0.8975 - val_loss: 0.1969 - val_accuracy: 0.9062 - 24s/epoch - 974ms/step  
Epoch 17/20  
25/25 - 24s - loss: 0.2395 - accuracy: 0.9050 - val_loss: 0.1840 - val_accuracy: 0.9141 - 24s/epoch - 970ms/step  
Epoch 18/20  
25/25 - 24s - loss: 0.2676 - accuracy: 0.9075 - val_loss: 0.2200 - val_accuracy: 0.9141 - 24s/epoch - 972ms/step  
Epoch 19/20  
25/25 - 24s - loss: 0.1824 - accuracy: 0.9388 - val_loss: 0.1876 - val_accuracy: 0.9531 - 24s/epoch - 970ms/step  
Epoch 20/20  
25/25 - 24s - loss: 0.2077 - accuracy: 0.9212 - val_loss: 0.1448 - val_accuracy: 0.9375 - 24s/epoch - 975ms/step
```

Gambar 5. Hasil uji coba skenario 2



Gambar 6. Grafik hasil uji coba skenario 2

4.3 Hasil Uji Coba Skenario 3

Pada skenario 3, citra dibagi menjadi dua dengan perbandingan 70:30 untuk data training dan data validasi. Jumlah citra yang digunakan adalah 3780 untuk data training, dan 1620 untuk data validasi. Citra akan di *rescale* sebesar $1/255$ dari nilai aslinya, rentangan ini berfungsi untuk mempermudah proses belajar dari jaringan syaraf tiruan. Selanjutnya skenario lain adalah ketika citra dirotasi sebesar 20 derajat, dilebarkan sebesar 20% dari ukuran aslinya, ditinggikan sebesar 20% dari ukuran aslinya, dipangkas sebesar 20% dari ukuran aslinya, diperbesar sebanyak 20%, dan jika citra di *flip* secara horizontal. Nilai piksel yang kosong pada citra akan diisi dengan nilai piksel terdekat. Uji coba pada skenario 1 menunjukkan hasil akurasi tertinggi yaitu 90% untuk data *training* dan 94% untuk data *validation*. Proses uji coba dapat dilihat pada Gambar 7 dan grafik akurasi dapat dilihat pada Gambar 8.

```
Epoch 1/20  
25/25 - 26s - loss: 1.9919 - accuracy: 0.4375 - val_loss: 0.9055 - val_accuracy: 0.6094 - 26s/epoch - 1s/step  
Epoch 2/20  
25/25 - 25s - loss: 0.8598 - accuracy: 0.6075 - val_loss: 0.8614 - val_accuracy: 0.6016 - 25s/epoch - 990ms/step  
Epoch 3/20  
25/25 - 24s - loss: 0.7831 - accuracy: 0.6464 - val_loss: 0.6398 - val_accuracy: 0.7422 - 24s/epoch - 961ms/step  
Epoch 4/20  
25/25 - 24s - loss: 0.8133 - accuracy: 0.6412 - val_loss: 0.6756 - val_accuracy: 0.6406 - 24s/epoch - 960ms/step  
Epoch 5/20  
25/25 - 25s - loss: 0.6751 - accuracy: 0.6988 - val_loss: 0.7117 - val_accuracy: 0.7031 - 25s/epoch - 987ms/step  
Epoch 6/20  
25/25 - 25s - loss: 0.7055 - accuracy: 0.7125 - val_loss: 0.5649 - val_accuracy: 0.7891 - 25s/epoch - 982ms/step  
Epoch 7/20  
25/25 - 25s - loss: 0.6210 - accuracy: 0.7287 - val_loss: 0.5457 - val_accuracy: 0.7891 - 25s/epoch - 980ms/step  
Epoch 8/20  
25/25 - 24s - loss: 0.5819 - accuracy: 0.7538 - val_loss: 0.5062 - val_accuracy: 0.8047 - 24s/epoch - 979ms/step  
Epoch 9/20  
25/25 - 24s - loss: 0.5709 - accuracy: 0.7650 - val_loss: 0.4483 - val_accuracy: 0.8203 - 24s/epoch - 977ms/step  
Epoch 10/20  
25/25 - 24s - loss: 0.5164 - accuracy: 0.7912 - val_loss: 0.3649 - val_accuracy: 0.8438 - 24s/epoch - 974ms/step  
Epoch 11/20  
25/25 - 24s - loss: 0.5126 - accuracy: 0.8018 - val_loss: 0.5739 - val_accuracy: 0.7031 - 24s/epoch - 949ms/step  
Epoch 12/20  
25/25 - 24s - loss: 0.4544 - accuracy: 0.8037 - val_loss: 0.6634 - val_accuracy: 0.6484 - 24s/epoch - 966ms/step  
Epoch 13/20  
25/25 - 24s - loss: 0.5458 - accuracy: 0.7966 - val_loss: 0.4063 - val_accuracy: 0.8125 - 24s/epoch - 940ms/step  
Epoch 14/20  
25/25 - 24s - loss: 0.4305 - accuracy: 0.8487 - val_loss: 0.4587 - val_accuracy: 0.7734 - 24s/epoch - 965ms/step  
Epoch 15/20  
25/25 - 23s - loss: 0.3740 - accuracy: 0.8394 - val_loss: 0.5140 - val_accuracy: 0.7578 - 23s/epoch - 939ms/step  
Epoch 16/20  
25/25 - 24s - loss: 0.5668 - accuracy: 0.8075 - val_loss: 0.3468 - val_accuracy: 0.8516 - 24s/epoch - 964ms/step  
Epoch 17/20  
25/25 - 24s - loss: 0.2800 - accuracy: 0.8963 - val_loss: 0.4099 - val_accuracy: 0.8516 - 24s/epoch - 966ms/step  
Epoch 18/20  
25/25 - 24s - loss: 0.3424 - accuracy: 0.8666 - val_loss: 0.5834 - val_accuracy: 0.7734 - 24s/epoch - 941ms/step  
Epoch 19/20  
25/25 - 24s - loss: 0.2757 - accuracy: 0.8913 - val_loss: 0.2473 - val_accuracy: 0.8984 - 24s/epoch - 960ms/step  
Epoch 20/20  
25/25 - 23s - loss: 0.2528 - accuracy: 0.9054 - val_loss: 0.1902 - val_accuracy: 0.9453 - 23s/epoch - 934ms/step
```

Gambar 7. Hasil uji coba skenario 3

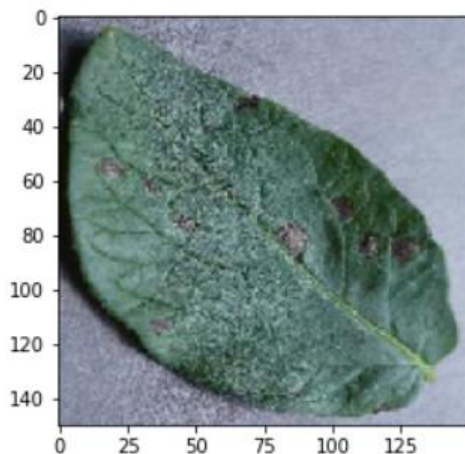


Gambar 8. Grafik hasil uji coba skenario 3

4.4 Identifikasi Citra

Metode convolutional neural network (CNN) memanfaatkan *layer* konvolusi dengan menggerakkan sebuah filter berukuran tertentu kedalam sebuah citra. Kemudian CNN akan memecah citra menjadi beberapa bagian kecil dengan *layer* konvolusi yang sama. Nilai dari konvolusi tersebut kemudian dimasukkan kedalam sebuah array baru, dimana array tersebut akan digunakan oleh jaringan syaraf untuk mengidentifikasi sebuah citra. Hasil identifikasi citra daun kentang berdasarkan hasil dari *train model* dapat dilihat pada Gambar 9.

Choose Files PotatoEarlyBlight3.JPG
• **PotatoEarlyBlight3.JPG**(image/jpeg) - 17949 bytes, last modified: 10/12/2019 - 100% done
Saving PotatoEarlyBlight3.JPG to PotatoEarlyBlight3.JPG
PotatoEarlyBlight3.JPG
Citra yang dipilih adalah citra dengan penyakit early blight



Gambar 9. Identifikasi citra

5. KESIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan dapat disimpulkan bahwa metode deep learning CNN dapat diimplementasikan untuk identifikasi citra penyakit pada daun kentang. Nilai akurasi tertinggi data *training* mencapai 93% dan akurasi tertinggi data *validation* mencapai 99% sehingga dapat disimpulkan bahwa metode deep learning CNN dapat mengidentifikasi citra penyakit pada daun kentang dengan baik.

DAFTAR PUSTAKA

- [1] L. Amatullah, I. Ein, and M. M. Santoni, "Identifikasi Penyakit Daun Kentang Berdasarkan Fitur Tekstur dan Warna Dengan Menggunakan Metode K-Nearest Neighbor," in *Seminar Nasional Mahasiswa Ilmu Komputer dan Aplikasinya (SENAMIKA)*, 2021, vol. 2, no. 1, pp. 783–791.
- [2] Felix, S. Faisal, T. F. M. Butarbutar, and P. Sirait, "Implementasi CNN dan SVM untuk Identifikasi Penyakit Tomat via Daun," *J. SIFO Mikroskil*, vol. 20, no. 2, pp. 117–134, 2019.
- [3] N. Ahmad and Iskandar, "Metode Forward Chaining untuk Deteksi Penyakit Pada Tanaman Kentang," *JINTECH J. Inf. Technol.*, vol. 1, no. 2, pp. 7–20, 2020, doi: 10.22373/jintech.v1i2.592.
- [4] A. J. Rozaqi, A. Sunyoto, and M. rudyanto Arief, "Deteksi Penyakit Pada Daun Kentang Menggunakan Pengolahan Citra dengan Metode Convolutional Neural Network," *Creat. Inf. Technol. J.*, vol. 8, no. 1, pp. 22–31, 2021, doi: 10.24076/citec.2021v8i1.263.
- [5] E. Rasywir, R. Sinaga, and Y. Pratama, "Analisis dan Implementasi Diagnosis Penyakit Sawit dengan Metode Convolutional Neural Network (CNN)," *Paradig. - J. Komput. dan Inform.*, vol. 22, no. 2, pp. 117–123, 2020, doi: 10.31294/p.v22i2.8907.
- [6] Felix, J. Wijaya, S. P. Sutra, P. W. Kosasih, and P. Sirait, "Implementasi Convolutional Neural Network Untuk Identifikasi Jenis Tanaman Melalui Daun," *J. SIFO Mikroskil*, vol. 21, no. 1, pp. 1–10, 2020.
- [7] A. Grosir, P. Toko, and R. Bungursari, "Klasifikasi Penyakit Tanaman pada Daun Apel dan Anggur Menggunakan Convolutional Neural Networks," vol. 4, no. 2, pp. 10–17, 2017.
- [8] F. N. Fiqri, "Deteksi Covid-19 Berdasarkan Citra X-Ray Menggunakan Convolutional Neural Network," Bogor, 2020.
- [9] I. Hasan, S. Suprayogi, and H. B. Dyah, "Klasifikasi Jenis Jerawat Menggunakan Convolutional Neural Networks," *eProceedings Eng.*, vol. 8, no. 1, pp. 358–372, 2021.
- [10] D. M. F. Shanti, N. Hidayat, and R. C. Wihandika, "Implementasi Metode F-KNN (Fuzzy K-Nearest Neighbor) Untuk Diagnosis Penyakit Anjing," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 2, no. 12, pp. 7401–7407, 2018.
- [11] Y. Yusdian, K. Karya, and R. Vaisal, "Pengaruh Dosis Pupuk Kandang Ayam Terhadap Pertumbuhan dan Hasil Tanaman Kentang (*Solanum tuberosum* L.) Varietas Granola," *Paspalum J. Ilm. Pertan.*, vol. 6, no. 2, pp. 98–102, 2018, doi: 10.35138/paspalum.v6i2.92.
- [12] Z. Y. G. Tirtana, L. Sulistyowati, and A. Cholil, "Eksplorasi jamur endofit pada tanaman kentang (*Solanum tuberosum* L) Serta Potensi Antagonismenya Terhadap *Phytophthora infestans* (Mont.) de Barry Penyebab Penyakit Hawar Daun Secara In Vitro," *J. Hama dan Penyakit Tumbuh.*, vol. 2, no. 1, pp. 91–101, 2014.
- [13] I. R. Sastrahidayat, *Tanaman Kentang dan Pengendalian Hama Penyakitnya*. Malang: Universitas Brawijaya Press, 2011.
- [14] F. Nainggolan, H. D. Hutahaean, and A. Gea, "Sistem Pakar Mendeteksi Penyakit Pada Tanaman Sayur Sawi Dengan Metode Bayes," *J. Armada Inform.*, vol. 2, no. 2, pp. 109–118, 2018, doi: 10.36520/jai.v2i2.31.
- [15] K. M. Gold *et al.*, "Hyperspectral Measurements Enable Pre-symptomatic Detection and Differentiation of Contrasting Physiological Effects of Late Blight and Early Blight in Potato," *Remote Sens.*, vol. 12, no. 2, p. 286, 2020, doi: 10.3390/rs12020286.
- [16] W. Xue, K. G. Haynes, and X. Qu, "Characterization of Early Blight Resistance in Potato Cultivars," *Plant Dis.*, vol. 103, no. 4, pp. 629–637, 2019, doi: 10.1094/PDIS-05-18-0794-RE.
- [17] G. J. Bu'ulölö, A. Jacobus, and F. D. Kambey, "Identification of Cataract Eye Disease Using Convolutional Neural Network," *J. Tek. Inform.*, vol. 16, no. 4, pp. 375–382, 2021.
- [18] C. Nisa, E. Y. Puspaningrum, and H. Maulana, "Penerapan Metode Convolutional Neural Network Untuk Klasifikasi Penyakit Daun Apel pada Imbalanced Data," in *Prosiding Seminar Nasional Informatika Bela Negara (SANTIKA)*, 2020, vol. 1, pp. 169–175.
- [19] R. A. Pangestu, B. Rahmat, and F. T. Anggraeny, "Implementasi Algoritma CNN untuk

Klasifikasi Citra Lahan dan Perhitungan Luas,” *Inform. dan Sist. Inf.*, vol. 1, no. 1, pp. 166–174, 2020.

- [20] J. Brownlee, *Data Preparation for Machine Learning: Data Cleaning, Feature Selection and Data Transforms In Python*. Machine Learning Mastery, 2020.

Biodata Penulis

Alang Mulya Lesmana, lahir di Jakarta pada 11 November 1999, Lulus dari SMKN 1 Cikarang Selatan pada tahun 2018. Saat ini sedang melanjutkan studi untuk mendapatkan gelar sarjana (S1) di Universitas Singaperbangsa Karawang.

Ronna Putri Fadhillah, lahir di Subang pada tahun 2000, Lulus dari SMAN 1 Ciasem pada tahun 2018. Saat ini sedang melanjutkan studi untuk mendapatkan gelar sarjana (S1) di Universitas Singaperbangsa Karawang

Chaerur Rozikin. lahir di Indramayu pada 20 Mei 1983. Lulus S1 pada tahun 2010 Teknik Informatika UIN Jakarta, lulus S2 pada tahun 2015 Ilmu Komputer, Institut Pertanian Bogor. Saat ini merupakan dosen di Universitas Singaperbangsa Karawang.