

PENERAPAN METODE *CONVOLUTIONAL NEURAL NETWORK* (CNN) DALAM APLIKASI PENDETEKSI PENYAKIT DAUN TANAMAN KENTANG BERBASIS ANDROID

Sonnya Ghandi, Imam Ma'ruf Nugroho, Yudhi Raymond Ramadhan

Teknik Informatika, Sekolah Tinggi Teknologi Wastukencana
Jalan Cikopak No.53, Mulyamekar, Kec.Babakancikao, Kab. Purwakarta, Jawa Barat
sonnyaghandi31@wastukencana.ac.id

ABSTRAK

Kentang, tumbuhan dikotil dari keluarga *Solanaceae*, merupakan bahan pokok penting dalam industri pangan. Namun, pertanian kentang sering menghadapi tantangan penyakit pada daun yang signifikan mengurangi hasil panen. Penelitian ini bertujuan mengembangkan aplikasi Android menggunakan *Convolutional Neural Network* (CNN) untuk membantu petani atau pengelola pertanian mengidentifikasi penyakit pada daun kentang. Menggunakan pendekatan CRISP-DM yang meliputi tahap *Business Understanding*, *Data Understanding*, *Data Preparation*, *Modeling*, *Evaluation*, dan *Deployment*, CNN digunakan untuk *supervised learning* dengan dataset gambar penyakit pada daun tanaman kentang. Model CNN dilatih selama 50 *epoch* dengan hasil yang sangat baik, menghasilkan *training accuracy* 99.69%, *validation accuracy* 99.65%, *training loss* 1.76%, dan *validation loss* 2.64%, menunjukkan kemampuan model dalam mengenali penyakit pada daun kentang dengan tingkat akurasi yang tinggi. Dalam evaluasi menggunakan 320 sampel data tes, model mencapai *accuracy* 99.69%, menegaskan keberhasilan aplikasi dalam deteksi dan klasifikasi penyakit daun tanaman kentang menggunakan CNN.

Kata kunci: *deep learning*, *convolutional neural network*, daun kentang

1. PENDAHULUAN

Kentang, sebagai tumbuhan dikotil dari keluarga *Solanaceae*, berkembang biak melalui umbi yang efisien di lingkungan dingin yang mendukung pertumbuhan optimal[1]. Selain menjadi sumber karbohidrat penting untuk kebutuhan pangan manusia[2], kentang juga berperan vital dalam industri pangan. Konsumsi kentang per kapita di Indonesia meningkat dari 2,547 kg pada 2020 menjadi 2,820 kg pada 2021, dan mencapai 3,167 kg pada 2022, dengan pertumbuhan 12,28% dari 2021 hingga 2022 [3]. Namun, meskipun memiliki peran yang signifikan, pertanian kentang sering menghadapi berbagai tantangan yang menyebabkan penurunan hasil produksi. Pada tahun 2022, hasil produksi tanaman kentang di Indonesia mencapai 1,5 juta ton, tetapi pada tahun 2023 terjadi penurunan produksi menjadi 1,2 juta ton[4].

Penurunan produksi kentang sebesar 80% disebabkan oleh penyakit yang menghambat pertumbuhan [1]. Jamur penyebab penyakit ini diduga berasal dari umbi-umbi bibit dan pertumbuhannya dipengaruhi oleh suhu lingkungan. Jamur yang dimaksud adalah jamur Patogen dan jamur *Alternaria Solani* (Aumatullah et al., 2021). Secara umum, terdapat dua jenis penyakit yang disebabkan oleh jamur yang menyerang tanaman kentang, yakni *early blight* dan *late blight* [1]

Gejala awal penyakit busuk daun (*Late Blight*) adalah bercak basah di tepi daun atau bagian tengahnya, yang meluas dan menyebabkan daun berubah warna menjadi coklat atau abu-abu. Gejala penyakit bercak kering (*Early Blight*) ditandai dengan bercak kering coklat berbentuk lingkaran di bagian

bawah daun [5]. Dengan mengamati gejala tersebut, petani dapat segera mengatasi penyakit daun yang membusuk. Namun, identifikasi gejala ini seringkali terkendala karena tidak semua petani memiliki pengetahuan yang memadai untuk mengenali gejala penyakit secara komprehensif.

Penanganan penyakit pada daun kentang telah mengalami perkembangan signifikan, melibatkan kontribusi dari bidang teknologi, terutama informatika. Teknik pengolahan gambar digital atau *image processing* diimplementasikan untuk mengidentifikasi penyakit pada tanaman kentang. Pemanfaatan *image processing* membantu pengelola pertanian menangani tanaman yang terkena penyakit atau menunjukkan gejala tidak normal secara lebih efektif dan efisien. Seiring kemajuan teknologi, banyak penelitian mengembangkan pengolahan gambar digital dalam konteks pertanian, baik untuk identifikasi penyakit maupun analisis hasil produksi.

Penelitian ini bertujuan mengembangkan aplikasi berbasis Android yang membantu petani atau pengelola pertanian mengidentifikasi penyakit pada daun kentang menggunakan data gambar daun kentang. Identifikasi penyakit dibagi menjadi tiga kategori: daun sehat, *Late Blight*, dan *Early Blight*. Penelitian ini menggunakan algoritma *Convolutional Neural Network* (CNN), sebuah metode dalam bidang *Deep Learning*, untuk melakukan identifikasi tersebut.

2. TINJAUAN PUSTAKA

2.1. *Cross Industry Standard Process for Data Mining* (CRISP-DM)

CRISP-DM (*Cross Industry Standard Process for Data Mining*) didirikan pada akhir 1990-an oleh

empat perusahaan utama: Integral Solutions Ltd. (perusahaan solusi data mining komersial), NCR (perusahaan database), DaimlerChrysler (produsen mobil), dan OHRA (perusahaan asuransi). CRISP-DM diakui sebagai proses standar yang digunakan untuk mengatasi masalah dalam praktik data mining di berbagai industri [6]. Dalam siklus pengembangannya, CRISP-DM diakui sebagai metodologi data mining yang paling komprehensif dalam memenuhi kebutuhan proyek industri. Metodologi ini telah menjadi yang paling umum digunakan dalam proyek analitik, data mining, dan ilmu data. CRISP-DM terdiri dari enam tahapan: *Business Understanding*, *Data Understanding*, *Data Preparation*, *Modeling*, *Evaluation*, dan *Deployment* [7].

2.2. Machine Learning

Machine learning adalah cabang ilmu komputer yang bertujuan mengembangkan algoritma untuk memungkinkan komputer belajar dari data yang ada. Dengan kata lain, komputer diajarkan memahami pola dan informasi dari data historis, sehingga dapat meningkatkan kinerjanya dalam menganalisis dan mengolah informasi yang diberikan. Oleh karena itu, *machine learning* sering disebut sebagai proses "belajar dari data," di mana komputer memperoleh pengetahuan atau keterampilan baru melalui pengalaman yang diperoleh dari data. Tujuan utamanya adalah menciptakan model atau program yang dapat mengoptimalkan performa dalam memahami dan menginterpretasikan data yang diberikan [8].

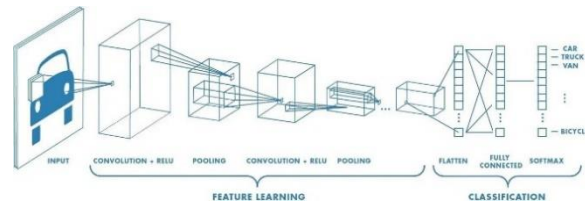
2.3. Deep Learning

Deep Learning merupakan metode dalam *machine learning* yang memproses informasi kompleks secara nonlinier dengan memanfaatkan berbagai lapisan. Teknik ini bertujuan untuk mengidentifikasi pola, mengekstrak ciri, dan melakukan klasifikasi melalui model komputasi yang dibangun dari struktur pemrosesan berlapis-lapis. Dengan memanfaatkan data pada berbagai tingkat abstraksi, deep learning memungkinkan pembelajaran yang mendalam dari informasi yang diberikan [9]. Sejak dikenal luas pada tahun 2006, *Deep Learning* menggunakan mekanisme *deep architecture of learning* atau *hierarchical learning*, yang mencakup proses estimasi parameter model untuk menyelesaikan tugas atau permasalahan tertentu [10]. *Deep Learning* menggunakan berbagai lapisan di antara lapisan masukan (*input layer*) dan lapisan keluaran (*output layer*), memungkinkan pemrosesan nonlinier melalui beberapa tahap untuk *feature learning* dan *pattern classification*[11].

2.4. Convolutional Neural Network

Convolutional Neural Network (CNN) adalah serangkaian lapisan pemrosesan yang menjalankan operasi konvolusi, bekerja secara paralel, dan terdiri dari beberapa elemen. Model ini terinspirasi oleh

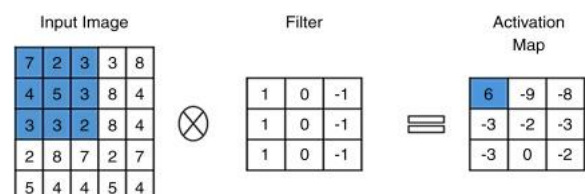
sistem saraf biologis [12]. Dalam domain *deep learning*, CNN digunakan untuk pengolahan gambar, menawarkan kinerja unggul dalam menangani masalah kompleks dan melakukan klasifikasi data gambar yang diwakili dalam bentuk matriks. CNN termasuk dalam kategori *supervised learning*, di mana model dilatih untuk mengidentifikasi pola antara *input* data dan label *output* [13].



Gambar 1. Convolutional Neural Network Layers

Gambar 1 menunjukkan bahwa inputan mengalami proses konvolusi pada segmen pembelajaran fitur (*feature learning*). Fitur, yang merupakan elemen-elemen dalam gambar yang diekstraksi oleh filter, dijadikan peta fitur (*feature map*). Dalam fase pembelajaran fitur, terdapat lapisan tersembunyi (*hidden layer*) yang mencakup lapisan konvolusi dan pooling maksimum (*max pooling*). Setelah segmen pembelajaran fitur, terdapat fase klasifikasi, di mana fitur diubah menjadi bentuk satu dimensi menggunakan teknik pelandaian (*flattening*). Fitur yang telah di-flatten kemudian diklasifikasikan melalui sejumlah *neuron*, mencapai *neuron* keluaran dengan jumlah yang setara dengan jumlah kelas dalam rangka klasifikasi. Berikut beberapa lapisan yang ada dimiliki metode CNN[14]:

- Convolution Layer*, adalah elemen kunci dalam arsitektur Jaringan Saraf Tiruan Konvolusional. Komponen ini terdiri dari kumpulan filter yang diterapkan melalui operasi konvolusi pada input yang disajikan. Operasi ini menghasilkan apa yang dikenal sebagai peta fitur. Dalam konteks lapisan konvolusional, setiap filter diwakili oleh matriks yang mengandung serangkaian nilai yang unik. Nilai-nilai ini, yang berfungsi mirip dengan bobot, akan mengalami proses pembaruan sepanjang fase pelatihan sesudah dilakukannya konvolusi. Operasi konvolusi itu sendiri melibatkan perkalian matriks antara filter dan lapisan konvolusional, dengan output dari operasi ini kemudian diakumulasikan untuk menghasilkan peta fitur yang diinginkan

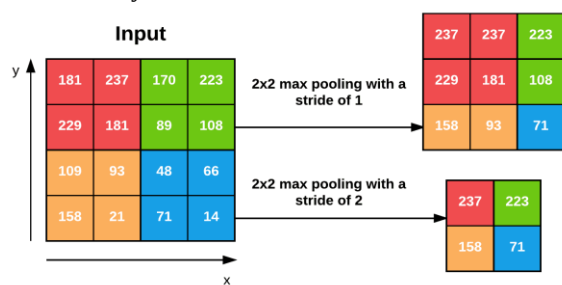


Gambar 2. Proses Konvolusi

Proses konvolusi pada gambar 2 melibatkan gambar input, filter, dan peta aktivasi. Gambar

input adalah matriks 2D yang merepresentasikan gambar asli, sedangkan filter adalah matriks kecil yang digunakan untuk memproses gambar input. Dalam contoh, filter dengan nilai-nilai tertentu diterapkan pada bagian tertentu dari gambar input dengan cara mengalikan elemen-elemen yang sesuai dan menjumlahkan hasilnya. Nilai yang diperoleh dari operasi ini ditempatkan di peta aktivasi. Proses ini diulangi dengan menggeser filter ke seluruh gambar input, menghasilkan matriks peta aktivasi yang merupakan representasi baru dari gambar yang menonjolkan fitur-fitur tertentu. Proses konvolusi ini umum digunakan dalam pengolahan gambar digital, khususnya dalam jaringan saraf konvolusional (CNN) untuk ekstraksi fitur dan deteksi pola.

- b. *Pooling Layer*, adalah sebuah komponen penting yang terletak antara lapisan konvolusi dalam arsitektur jaringan saraf tiruan. Fungsi utama dari *layer* ini adalah untuk melakukan pengurangan volume pada keluaran dari peta fitur, yang dapat menghasilkan percepatan komputasi pada lapisan berikutnya



Gambar 3. Proses *Pooling*

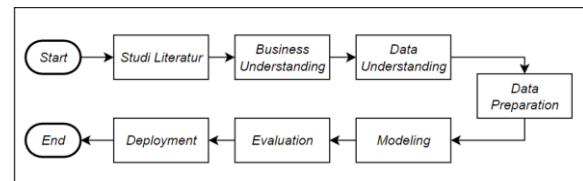
Gambar 3 menunjukkan proses max pooling, yaitu teknik dalam jaringan saraf konvolusional (CNN) untuk mengurangi dimensi data dan mengekstrak fitur penting. Dalam contoh ini, matriks input berukuran 4x4 diproses dengan *pooling window* berukuran 2x2. Pada *max pooling*, nilai maksimal dari setiap sub-matriks 2x2 diambil sebagai hasilnya. Terdapat dua cara yang ditunjukkan: dengan stride 1, di mana *pooling window* bergeser satu langkah setiap kali sehingga menghasilkan matriks keluaran yang lebih besar, dan dengan stride 2, di mana *pooling window* bergeser dua langkah setiap kali, menghasilkan matriks keluaran yang lebih kecil.

- c. *Fully Connected Layer*, adalah struktur yang menghubungkan setiap *neuron* aktivasi dari lapisan sebelumnya secara langsung dengan setiap *neuron* pada lapisan berikutnya. Perbedaan mendasar antara lapisan terhubung penuh dan lapisan konvolusi adalah bahwa dalam lapisan konvolusi, *neuron-neuron* hanya terhubung dengan wilayah tertentu dari input, sementara dalam lapisan terhubung penuh, setiap *neuron* terhubung dengan semua *neuron* pada lapisan sebelumnya. Pada titik ini, semua fitur yang telah dipelajari oleh jaringan dari gambar akan

digunakan untuk melakukan klasifikasi. Lapisan terhubung penuh dapat dianggap sebagai lapisan dari jaringan saraf tiruan tradisional (*multilayer perceptrons*) yang digunakan untuk mengklasifikasikan gambar berdasarkan fitur-fitur yang telah diidentifikasi oleh lapisan sebelumnya

3. METODE PENELITIAN

Pada penelitian ini ada beberapa tahapan penyelesaian dalam melakukan klasifikasi penyakit daun pada tanaman kentang seperti yang ditampilkan pada gambar 3.



Gambar 4. Kerangka penelitian

Gambar 4 menunjukkan bahwa pengembangan model pembelajaran dalam pada penelitian ini menerapkan kerangka kerja CRISP-DM dan ditambah satu tahapan lainnya, yang dibagi menjadi 7 tahap. Tahap-tahap tersebut dimulai dengan Studi Literatur, *Business Understanding*, diikuti oleh *Data Understanding*, kemudian berlanjut ke *Data Preparation*, *Modeling*, *Evaluation*, dan berakhir dengan *Deployment*.

3.1. Studi Literatur

Berdasarkan latar belakang masalah yang telah dirumuskan melalui analisis data dan dukungan dari jurnal serta penelitian terdahulu, dapat disimpulkan bahwa tujuan dari penelitian ini adalah untuk mengembangkan model yang mampu mengenali gambar daun tanaman kentang untuk diklasifikasikan berdasarkan kondisi kesehatannya. Klasifikasi gambar ini menggunakan metode *Convolutional Neural Network* (CNN) dalam pengembangan aplikasi pendeteksi penyakit daun tanaman kentang berbasis Android.

3.2. Business Understanding

Penelitian ini bertujuan untuk mengembangkan sebuah aplikasi seluler yang mampu mengklasifikasikan penyakit pada daun tanaman kentang menggunakan teknik pembelajaran terarah dengan memanfaatkan algoritma *Convolutional Neural Network*. Aplikasi ini akan mengidentifikasi penyakit pada daun tanaman kentang dengan mengambil gambar melalui kamera ponsel, dan selanjutnya akan menghasilkan output berupa tipe penyakit berdasarkan gambar yang diperoleh.

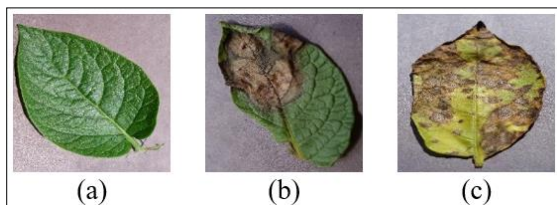
3.3. Data Understanding

Dataset yang digunakan dalam penelitian ini terdiri dari gambar penyakit pada daun tanaman kentang, yang diakses dari situs *Kaggle*.

Tabel 1. *Dataset Class*

Class	Definisi	Sumber
Healthy	Gambar daun tanaman kentang yang sehat	https://www.kaggle.com/datasets/abdallahalidev/plantvillage-dataset
Late Blight	Gambar daun tanaman kentang yang terinfeksi penyakit busuk daun	
Early Blight	Gambar daun tanaman kentang yang terinfeksi penyakit bercak kering	

Pada Tabel 1 dijelaskan Gambar dalam dataset ini mencakup tiga *class* atau label berbeda. Gambar dalam *dataset* tersebut telah diklasifikasikan secara manual oleh para ahli, memastikan bahwa setiap gambar memiliki label yang akurat dan sesuai dengan kondisi sebenarnya. Proses klasifikasi manual ini sangat penting untuk memastikan kualitas dan keandalan data yang digunakan dalam pelatihan model, karena label yang akurat memungkinkan model belajar lebih efektif dan menghasilkan prediksi yang lebih tepat. Dengan adanya klasifikasi manual oleh para ahli, dataset ini menjadi sumber data yang sangat berharga untuk pengembangan dan evaluasi model, khususnya dalam mendeteksi dan mengklasifikasikan penyakit pada daun tanaman kentang.

Gambar 5. (a) Daun sehat (b) *Late blight* (c) *Early blight*

Pada Gambar 5 (a) menampilkan daun kentang yang sehat, tanpa tanda-tanda penyakit. Gambar 2 (b) ditampilkan daun tanaman kentang yang terkena penyakit busuk daun (*late blight*). Gambar 2 (c) menunjukkan daun kentang yang terkena penyakit bercak daun (*early blight*), terlihat adanya bercak-bercak kecokelatan di tengah daun yang merusak permukaan daun ditandai dengan perubahan warna menjadi kecokelatan pada tepi daun, yang semakin meluas seiring waktu.

3.4. Data Preparation

Sebelum dilakukan pelatihan model, dataset perlu mengalami proses data augmentation. Augmentasi data adalah metode yang memungkinkan penambahan variasi data untuk pelatihan model tanpa perlu mengumpulkan data baru. Beberapa teknik augmentasi data termasuk pemotongan (*cropping*), penambahan tepi (*padding*), dan pembalikan gambar secara horizontal (*horizontal flipping*) [15].

3.5. Modeling

Dalam penelitian ini, algoritma *Convolutional Neural Network* (CNN) digunakan untuk melatih model menggunakan dataset gambar. Setiap gambar yang menjadi masukan memiliki resolusi 256x256 piksel dan menggunakan tiga kanal warna yaitu Merah (*Red*), Hijau (*Green*), dan Biru (*Blue*). Proses pelatihan dilakukan secara iteratif hingga model mencapai tingkat akurasi yang optimal.

Pada tahap *modeling* ini ada dua tahap utama yaitu pembuatan model dan penentuan parameter. Pembuatan model yang dilakukan yaitu membuat model *sequential* dengan menggunakan konfigurasi arsitektur dengan 6 *convolution layer*, 6 *pooling layer*, dan 1 *hidden layer*. Penentuan parameter yang akan digunakan adalah sebagai berikut.

Tabel 2 Parameter *modeling*

Parameter	Epoch
Parameter 1	30
Parameter 2	40
Parameter 3	50

Tabel 2 menjelaskan beberapa parameter yang memiliki *epoch* yang berbeda.

3.6. Evaluation

Pada tahap ini, model yang telah dilatih dievaluasi dengan melihat grafik tingkat akurasi, yang mencakup *training accuracy* dan *validation accuracy*, serta grafik tingkat *error*, yang mencakup *training loss* dan *validation loss*. Jika nilai-nilai tersebut menunjukkan indikasi *overfitting* atau *underfitting*, maka perancangan ulang dan pelatihan model kembali akan dilakukan. *Overfitting* terjadi ketika model menunjukkan kinerja sangat baik pada data pelatihan namun menurun pada data uji karena model terlalu menyesuaikan diri dengan data pelatihan, kehilangan kemampuan untuk menggeneralisasi. Sebaliknya, *underfitting* terjadi ketika model tidak mampu menangkap pola pada data pelatihan dengan baik, sehingga kinerjanya buruk pada data pelatihan dan data uji, biasanya karena model terlalu sederhana untuk kompleksitas data [16].

Tahap kedua dari evaluasi melibatkan pembuatan *confusion matrix* untuk model yang telah dibangun. *Confusion matrix* adalah tabel yang menunjukkan jumlah data uji yang diklasifikasikan dengan benar dan jumlah data uji yang diklasifikasikan secara salah [17].

3.7. Deployment

Setelah model melewati tahap pemodelan dan evaluasi, model tersebut siap dikonversi ke format *.tflite* menggunakan *TFLiteConverter* yang tersedia dalam *library TensorFlow*. Langkah selanjutnya adalah melakukan *deployment* model yang telah dikonversi ke dalam aplikasi mobile sehingga aplikasi dapat diunduh dan digunakan oleh siapapun yang memiliki *smartphone* dengan sistem operasi Android.

4. HASIL DAN PEMBAHASAN

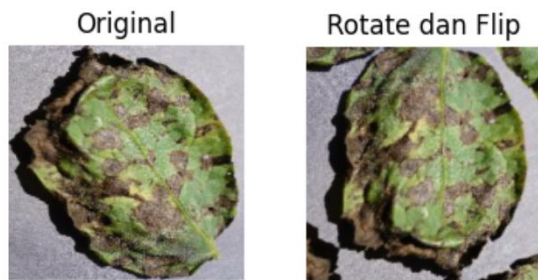
4.1. Data Preparation

Dataset yang diperoleh dari situs *Kaggle* akan mengalami *augmentasi* data sebelum memasuki proses pemodelan. Berikut adalah *augmentasi* data yang digunakan dalam penelitian ini:

Pertama, *split dataset*. Dalam tahap ini, dataset dibagi menjadi dua bagian, yaitu 80% untuk data pelatihan, 10% untuk data validasi dan 10% untuk data uji.

Kedua, *resize* dan *rescale*. Pada tahap ini gambar akan dilakukan *rescale* sehingga gambar memiliki ukuran 256 piksel.

Ketiga, *rotation* dan *flip*. Pada tahap ini dilakukan rotasi sebesar 20% dan membalik gambar secara acak antara *horizontal* dan *vertical*.



Gambar 6. Hasil proses *rotate* dan *flip*

Gambar 6 merupakan gambar sebelum dan sesudah dilakukan *rotation* dan *flip*.

4.2. Modeling

Modeling terdiri dari dua bagian utama yaitu proses pembuatan model dan penentuan parameter model. Proses pembuatan model melibatkan tahapan-tahapan untuk membangun model, sedangkan penentuan parameter model adalah tahap di mana parameter-parameter model ditentukan dan dibandingkan untuk menilai performa mana yang lebih baik.

Tahap pertama pembuatan model diawali dengan membuat model *sequential*, model *sequential* yang diberikan terdiri dari 6 lapisan *convolution* dan 6 *pooling* yang digunakan untuk ekstraksi fitur dari gambar input, diikuti oleh 1 lapisan *fully connected* untuk klasifikasi. Model ini memiliki beberapa lapisan *convolution* (Conv2D) dengan 32 dan 64 *filter* yang menggunakan fungsi aktivasi *ReLU* untuk menangkap fitur spasial penting. Setiap lapisan konvolusi diikuti oleh lapisan *pooling* (*MaxPooling2D*) yang mengurangi dimensi spasial untuk mengurangi jumlah parameter dan mencegah *overfitting*. Setelah serangkaian lapisan konvolusi dan *pooling*, lapisan *Flatten* digunakan untuk meratakan *output* sebelum dimasukkan ke dalam lapisan *dense* (*Dense*) yang terdiri dari 64 *neuron* dengan aktivasi *ReLU*. Terakhir, lapisan *output* menggunakan fungsi aktivasi *Softmax* untuk menghasilkan probabilitas dari masing-masing *class* target dalam tugas klasifikasi. Selanjutnya yaitu proses kompilasi model, Model ini dikompilasi menggunakan *optimizer Adam* untuk pembaruan

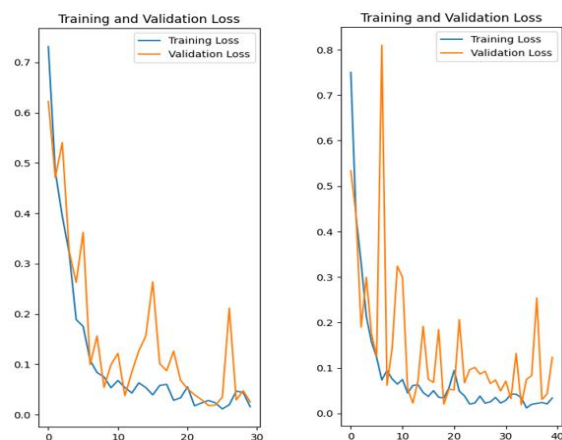
parameter, dengan fungsi yang digunakan adalah *loss SparseCategoricalCrossentropy* karena dataset memiliki lebih dari dua *class*, serta *from_logits=False* karena output model telah melalui fungsi aktivasi *softmax*. Metrik evaluasi yang dipilih adalah *accuracy* untuk mengukur performa model selama pelatihan dan pengujian.

Tahap kedua penentuan parameter model, pada saat melakukan training, model harus menentukan parameter-parameter yang akan mempengaruhi seberapa baik performa model setelah selesai dilatih. Salah satu parameter penting yang perlu ditentukan adalah jumlah *epoch*. Penentuan parameter ini bertujuan untuk membandingkan performa model berdasarkan berbagai nilai parameter dan memilih parameter yang memberikan performa terbaik untuk digunakan dalam model

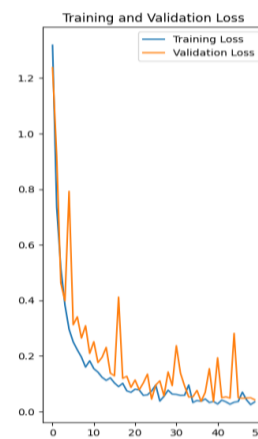
Tabel 3. Perbandingan *epoch* proses *training model*

Epoch	Training		Validation	
	Accuracy	Loss	Accuracy	Loss
30	0.9945	0.0168	0.9896	0.0219
40	0.9919	0.0232	0.9896	0.0366
50	0.9969	0.0176	0.9965	0.0264

Pada Tabel 3 terlihat bahwa proses *training* dengan *epoch* 50 memiliki akurasi terbesar. Untuk gambaran yang lebih jelas bisa dilihat Gambar 6 dan 7 berikut ini.



Gambar 7. Grafik *loss* 30 dan 40



Gambar 8. Grafik *loss* 50

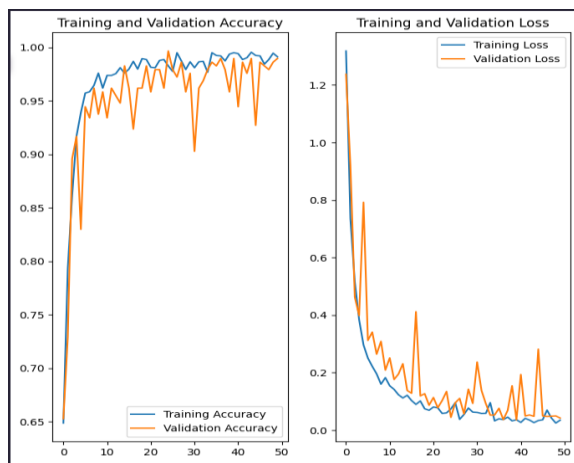
Dari Gambar 7, 8 dan Tabel 2 menyatakan bahwa *model* dengan menggunakan 50 *epoch* memiliki hasil yang lebih baik, dengan nilai *training accuracy* sebesar 99.69%, *validation accuracy* sebesar 99.65%, *training loss* sebesar 1.76% dan *validation loss* sebesar 2.64%.

4.3. Evaluation

Tahap evaluasi yang dilakukan dalam penelitian ini meliputi analisis grafik dari *model* yang telah dilatih dan pengamatan hasil dari *confusion matrix*. Berikut ini hasil evaluasi model yang menggunakan 50 *epoch*.

```
74/74 [=====] - 0s 23ms/step - loss: 0.0176 - accuracy: 0.9969
9/9 [=====] - 0s 29ms/step - loss: 0.0264 - accuracy: 0.9965
Training Loss: 0.017597202211618423
Training Accuracy: 0.996874988079071
Val Loss: 0.02637781947851181
Val Accuracy: 0.9965277910232544
```

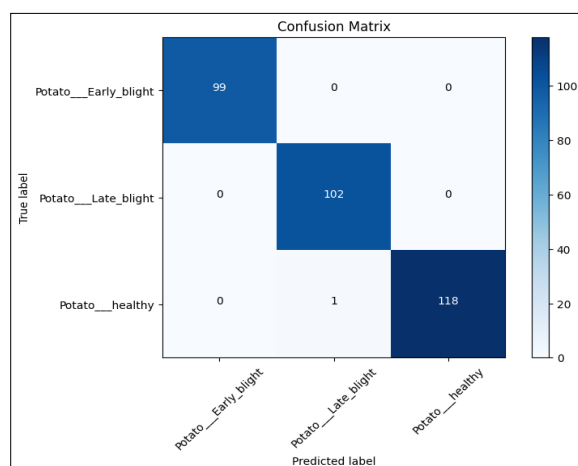
Gambar 9. Nilai *evaluation model*



Gambar 10. Grafik *training model*

Dari gambar 9 dan 10 menyatakan bahwa model memiliki akurasi yang bagus yaitu dengan nilai *training accuracy* sebesar 99.69%, *validation accuracy* sebesar 99.65%, *training loss* sebesar 1.76% dan *validation loss* sebesar 2.64%.

Selanjutnya evaluasi model dengan *confusion matrix*. Berikut ini gambar 10 hasil *confusion matrix*.



Gambar 11. *Confusion matrix*

Berikut ini perhitungan akurasi dari *confusion matrix model* pada Gambar 11.

$$\begin{aligned} \text{Akurasi} &= \frac{\text{Jumlah true positif}}{\text{Jumlah data test}} \\ &= \frac{99 + 102 + 118}{320} \\ &= 99.69\% \end{aligned} \quad (1)$$

Maka, hasil akurasi model dari jumlah data test sebanyak 320 data gambar adalah 99.69%.

4.4. Deployment

Melakukan penerapan dan deployment model yang telah dibangun ke dalam aplikasi *mobile*, sehingga aplikasi dapat diunduh dan dapat digunakan oleh siapapun yang menggunakan *smartphone* dengan sistem operasi Android. Berikut ini adalah tampilan dari aplikasi *mobile* yang telah berhasil dikembangkan dan dibangun dengan berbagai fitur yang dirancang untuk memenuhi kebutuhan pengguna secara optimal yang bernama “PotatoGuard”.



Gambar 12. Halaman *onboarding*

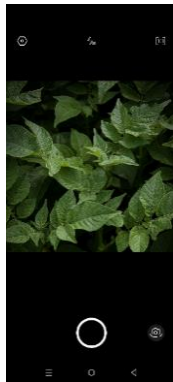
Pertama, Gambar 12 adalah halaman *onboarding*. Halaman ini merupakan halaman pertama yang muncul sebelum ke halaman utama aplikasi.



Gambar 13. Halaman utama

Kedua, Gambar 13 halaman utama. Menampilkan semua jenis klasifikasi daun *kentang*,

dan terdapat tombol untuk membuka halaman kamera. Berikut gambar 4.8 halaman utama



Gambar 14. Halaman kamera

Ketiga, Gambar 14 halaman *kamera*. Berfungsi untuk mengambil gambar daun kentang yang akan dideteksi dan diklasifikasi



Gambar 15. Halaman klasifikasi

Keempat, Gambar 15 halaman klasifikasi. Fitur utama dalam *aplikasi* ini, fungsinya yaitu menerima inputan gambar dari halaman kamera lalu kemudian diklasifikasi jenis penyakitnya



Gambar 16. Halaman detail penyakit

Kelima, Gambar 16 halaman detail penyakit. Menampilkan detail penyakit yang sudah diklasifikasi dari halaman klasifikasi, juga berisi mengenai penyebab, gejala, dan cara pengendalian dari penyakitnya

5. KESIMPULAN DAN SARAN

Hasil penelitian menunjukkan bahwa metode *Convolutional Neural Network* (CNN) berhasil diimplementasikan dengan baik pada aplikasi pendeteksi penyakit daun tanaman kentang berbasis Android, dengan hasil deteksi dan klasifikasi yang efektif. Model yang dilatih dengan 50 *epoch* menunjukkan performa yang sangat baik dengan nilai *training accuracy* sebesar 99.69%, *validation accuracy* sebesar 99.65%, *training loss* sebesar 1.76%, dan *validation loss* sebesar 2.64%. Evaluasi dengan 320 sampel data tes menunjukkan akurasi sebesar 99.69%. Untuk penelitian selanjutnya, disarankan untuk menambah jenis penyakit daun tanaman kentang pada ubi kentang. Selain itu, pengembangan sistem berbasis *Internet of Things* (IoT) yang mengintegrasikan data sensor lingkungan seperti kelembaban dan suhu dapat membantu dalam deteksi penyakit, karena kondisi lingkungan juga mempengaruhi kesehatan tanaman.

DAFTAR PUSTAKA

- [1] A. Fuadi dan A. Suharso, "Perbandingan Arsitektur Mobilenet Dan Nasnetmobile Untuk Klasifikasi Penyakit Pada Citra Daun Kentang," *JIPI (Jurnal Ilm. Penelit. dan Pembelajaran Inform.*, vol. 7, no. 3, hal. 701–710, 2022, doi: 10.29100/jipi.v7i3.3026.
- [2] E. Auliatusahra, "ANALISIS PERAN BIOTEKNOLOGI REKAYASA GENETIKA UNTUK PENINGKATAN KUALITAS PADA TOMAT DAN KENTANG," *Humanit. J. Humaniora, Sos. dan Bisnis*, vol. 2, no. 1, hal. 113–123, 2024.
- [3] Kementerian Pertanian, "Statistik Konsumsi Pangan Tahun 2022," *Pus. Data dan Sist. Inf. Pertanian, Kementerian Pertan. Republik Indones.*, hal. 1–132, 2022.
- [4] Badan Pusat Statistik, "Produksi Tanaman Sayuran, 2021-2023," bps.go.id. Diakses: 26 Juni 2024. [Daring]. Tersedia pada: <https://www.bps.go.id/id/statistics-table/2/NjEjMg==/produksi-tanaman-sayuran.html>
- [5] A. J. Rozaqi, A. Sunyoto, dan M. rudyanto Arief, "Deteksi Penyakit Pada Daun Kentang Menggunakan Pengolahan Citra dengan Metode Convolutional Neural Network," *Creat. Inf. Technol. J.*, vol. 8, no. 1, hal. 22, 2021, doi: 10.24076/citec.2021v8i1.263.
- [6] A. Pambudi, "Penerapan Crisp-Dm Menggunakan Mlr K-Fold Pada Data Saham Pt. Telkom Indonesia (Persero) Tbk (Tlkm) (Studi Kasus: Bursa Efek Indonesia Tahun 2015-2022)," *J. Data Min. dan Sist. Inf.*, vol. 4, no. 1, hal. 1, 2023, doi: 10.33365/jdmsi.v4i1.2462.
- [7] D. Kurniawan dan M. Yasir, "Optimization Sentimen Analysis using CRISP-DM and Naive Bayes Methods Implemented on Social Media,"

- Cybersp. J. Pendidik. Teknol. Inf.*, vol. 6, no. 2, hal. 74, 2022, doi: 10.22373/cj.v6i2.12793.
- [8] A. D. Sidik dan A. Ansawarman, "Prediksi Jumlah Kendaraan Bermotor Menggunakan Machine Learning," *Formosa J. Multidiscip. Res.*, vol. 1, no. 3, hal. 559–568, 2022, doi: 10.55927/fjmr.v1i3.745.
- [9] Radikto, D. I. Mulyana, M. A. Rofik, dan Mo. Z. Zakaria, "Klasifikasi Kendaraan pada Jalan Raya menggunakan Algoritma Convolutional Neural Network (CNN)," *J. Pendidik. Tambusai*, vol. 6, no. 1, hal. 1668–1679, 2022.
- [10] M. Z. Alom *et al.*, "A state-of-the-art survey on deep learning theory and architectures," *Electron.*, vol. 8, no. 3, 2019, doi: 10.3390/electronics8030292.
- [11] Muhammad Haris Diponegoro, Sri Suning Kusumawardani, dan Indriana Hidayah, "Tinjauan Pustaka Sistematis: Implementasi Metode Deep Learning pada Prediksi Kinerja Murid," *J. Nas. Tek. Elektro dan Teknol. Inf.*, vol. 10, no. 2, hal. 131–138, 2021, doi: 10.22146/jnteti.v10i2.1417.
- [12] M. H. Romario, E. Ihsanto, dan T. M. Kadarina, "Sistem Hitung dan Klasifikasi Objek dengan Metode Convolutional Neural Network," *J. Teknol. Elektro*, vol. 11, no. 2, hal. 108, 2020, doi: 10.22441/jte.2020.v11i2.007.
- [13] N. H. Harani, C. Prianto, dan M. Hasanah, "Deteksi Objek Dan Pengenalan Karakter Plat Nomor Kendaraan Indonesia Menggunakan Metode Convolutional Neural Network (CNN) Berbasis Python," *J. Tek. Inform.*, vol. 11, no. 3, hal. 47–53, 2019, [Daring]. Tersedia pada: <https://ejurnal.ulbi.ac.id/index.php/informatika/article/view/658>
- [14] F. S. Anam, M. R. Muttaqin, dan Y. R. Ramadhan, "Klasifikasi Penyakit Pada Daun dan Buah Jambu Menggunakan Convolutional Neural Network," *JOINTECS (Journal Inf. Technol. Comput. Sci.)*, vol. 8, no. 3, hal. 115, 2023, doi: 10.31328/jointecs.v8i3.4823.
- [15] J. Sanjaya dan M. Ayub, "Augmentasi Data Pengenalan Citra Mobil Menggunakan Pendekatan Random Crop, Rotate, dan Mixup," *J. Tek. Inform. dan Sist. Inf.*, vol. 6, no. 2, hal. 311–323, 2020, doi: 10.28932/jutisi.v6i2.2688.
- [16] Rima Dias Ramadhani, A. Nur Aziz Thohari, C. Kartiko, A. Junaidi, T. Ginanjar Laksana, dan N. Alim Setya Nugraha, "Optimasi Akurasi Metode Convolutional Neural Network untuk Identifikasi Jenis Sampah," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 5, no. 2, hal. 312–318, 2021, doi: 10.29207/resti.v5i2.2754.
- [17] D. Normawati dan S. A. Prayogi, "Implementasi Naïve Bayes Classifier Dan Confusion Matrix Pada Analisis Sentimen Berbasis Teks Pada Twitter," *J. Sains Komput. Inform.*, vol. 5, no. 2, hal. 697–711, 2021.