

KLASIFIKASI PENYAKIT TANAMAN APEL DARI CITRA DAUN DENGAN CONVOLUTIONAL NEURAL NETWORK

Andi Asrafil Ardan Paliwang¹⁾, M Ridwan Dwi Septian²⁾, Margi Cahyanti³⁾, dan Ericks Rachmat Swedia⁴⁾

^{1,3,4} Jurusan Teknik Informatika, Fakultas Teknologi Informasi, Universitas Gunadarma

² Jurusan Sistem Informasi, Fakultas Ilmu Komputer, Universitas Gunadarma

^{1,2,3,4} Jalan Margonda Raya No.100, Pondok Cina, Depok

E-mail : andiasrafil78@student.gunadarma.ac.id¹⁾, ridwandwiseptian@staff.gunadarma.ac.id²⁾, margi@staff.gunadarma.ac.id³⁾, ericks_rs@staff.gunadarma.ac.id⁴⁾

ABSTRAK

Tanaman Apel merupakan buah tahunan yang berasal dari daerah Asia Barat dengan iklim sub tropis. Di Indonesia tanaman Apel ditanam sejak tahun 1934 hingga saat ini. Tanaman Apel dapat tumbuh dan berbuah baik di daerah dataran tinggi. Para petani biasanya melakukan pencegahan penyakit atau hama dengan melakukan penyemprotan setiap 1 – 2 minggu sekali dengan dosis ringan. Pencegahan ini agar penyakit/hama dapat segera ditanggulangi dan baik jika dilakukan pada pagi atau sore hari. Terkadang petani juga membutuhkan seorang pakar dalam menentukan jenis hama/penyakit pada tanaman Apel agar dapat memberikan solusi terbaik. Oleh karena itu diperlukan adanya pendekatan *digital* agar dapat mengenali beragam jenis hama/penyakit tanaman Apel dengan cepat dan mudah. Teknologi *Deep learning*, merupakan salah satu bidang ilmu baru dalam *Machine Learning* dan berkembang dengan pesat. *Deep learning* memiliki kemampuan yang baik dalam *Computer Vision*, salah satunya yaitu *image classification* atau klasifikasi objek pada citra. Metode yang dapat digunakan dalam melakukan klasifikasi citra ini yaitu *Convolutional Neural Network (CNN)*. Berdasarkan hasil uji coba, aplikasi berhasil diimplementasikan dengan baik menggunakan *framework dart* berbasis android dengan hasil *final test accuracy* yang diperoleh yaitu didapat keakuratan akurasi sebesar 97,1%.

Kata Kunci: *Apel, Citra, Daun, CNN, Image, Klasifikasi.*

1. PENDAHULUAN

Apel merupakan tanaman buah tahunan yang berasal dari daerah Asia Barat dengan iklim sub tropis. Di Indonesia Apel telah ditanam sejak tahun 1934 hingga saat ini. Apel dapat tumbuh dan berbuah baik di daerah dataran tinggi. Salah satu sentra produksi Apel di Indonesia adalah Batu dan Poncokusumo, Kab. Malang (Permana dkk., 2015).

Di Kota Batu, kegiatan budidaya Apel berlangsung hampir setiap tahunnya dengan dua kali masa panen. Pengendalian hama dan penyakit merupakan salah satu faktor terpenting dalam perkembangan tanaman Apel. Para petani biasanya melakukan pencegahan dengan melakukan penyemprotan setiap 1 – 2 minggu sekali dengan dosis ringan. Pencegahan ini agar hama dapat segera ditanggulangi dan baik dilakukan di pagi atau sore hari. Terkadang petani juga membutuhkan seorang pakar dalam menentukan jenis hama dan penyakit pada tanaman Apel agar dapat memberikan solusi terbaik (Permana dkk., 2015).

Deep learning merupakan salah satu cabang dari *machine learning*. Model *deep learning* dapat mempelajari komputasinya sendiri dengan menggunakan otaknya sendiri. *Deep learning* dirancang untuk terus menganalisis data seperti pada otak manusia dalam mengambil keputusan. Agar kemampuan *deep learning* semakin pintar, maka *deep learning* menggunakan

algoritma *Artificial Neural Network (ANN)* yang terpisah dari jaringan biologis otak manusia.

Salah satu jenis *neural network* yang biasa digunakan pada data *image* adalah CNN. Karena dalamnya tingkat jaringan maka CNN termasuk dalam jenis *deep neural network* dan sering digunakan dalam data citra. Ada dua metode yang dimiliki oleh CNN, yaitu klasifikasi menggunakan *feedforward* dan tahap pembelajaran menggunakan *backpropagation*.

Pada klasifikasi citra, CNN mendapat masukan atau citra masukan untuk diproses ke dalam dan diklasifikasi ke kategori tertentu. Perbedaan CNN dan ANN adalah pada arsitektur tambahan pada CNN yang dioptimalkan untuk fitur yang ada pada citra masukan. Ada beberapa komponen utama yang ada dalam CNN yaitu di antaranya adalah *convolutional layer*, *pooling layer*, *fully connected layer* dan *dropout* (Sharif Razavian dkk., 2014).

2. RUANG LINGKUP

Permasalahan pada penelitian ini dapat dilihat sebagai berikut :

1. Penerapan metode CNN dalam klasifikasi penyakit tanaman apel dengan menggunakan bahasa pemrograman Python 3.6 dan Google Colab.
2. Dataset yang digunakan diambil dari *keggle* yang menyediakan dataset.

3. Penyakit yang digunakan untuk klasifikasi hanya 3 penyakit, yaitu *scab*, *blackrot*, dan *rust*.
4. Aplikasi ini hanya dapat dijalankan pada *smartphone* yang memiliki sistem operasi Android.
5. *Output* berupa hasil klasifikasi jenis penyakit tanaman Apel.

3. BAHAN DAN METODE

Pada bagian ini membahas penggunaan metode yang terkait dalam penelitian, di antaranya :

3.1 Tanaman Apel

Apel dalam ilmu botani disebut *Malus sylvestris* Mill. Apel merupakan tanaman buah tahunan yang berasal dari daerah Asia Barat dengan iklim sub tropis. Di Indonesia Apel telah ditanam sejak tahun 1934 hingga saat ini (Suhardjo, H. L., Deaton, B. J., & Driskel, 1985). Menurut sistematika, tanaman Apel termasuk dalam, divisio : *Spermatophyta*, subdivisio : *Angiospermae*, klas : *Dicotyledonae*, ordo : *Rosales*, famili : *Rosaceae*, genus : *Malus*, spesies : *Malus sylvestris* Mill

Dari spesies *malus sylvestris mill* ini, ada bermacam-macam varietas yang mempunyai ciri-ciri atau karakteristik khas tersendiri. Beberapa varietas Apel yang termasuk unggulan antara lain: Rome Beauty, Manalagi, Anna, Princess Noble dan Wangli/Lali jiwo. Apel memiliki kandungan banyak vitamin C serta vitamin B. Selain itu para pelaku diet menjadi pilihan untuk melakukan diet sebagai ganti makanan substitusi (Ernawati, 2018). Gangguan pada tanaman Apel disebabkan oleh hama maupun penyakit. Beberapa hama dan penyakit yang menyerang tanaman Apel seperti pada tabel 1.

Tabel 1. Jenis Hama

Jenis Hama
Kutu Hijau (<i>Aphis pomi</i> Geer)
Tungau, Spinder mite, Cambuk merah (<i>Panonychus Ulmi</i>)
Trips
Ulat daun
Serangga penghisap daun
Ulat daun hitam
Lalat buah

3.2 Deep learning

Deep learning merupakan metode pembelajaran representasi yang memungkinkan model komputasional yang tersusun dari banyak lapisan pengolahan untuk belajar representasi data dengan banyak tingkat abstraksi. Metode ini secara signifikan meningkatkan unjuk kerja pada pengenalan percakapan, pengenalan obyek visual, deteksi obyek dan lain sebagainya (LeCun dkk., 2015).

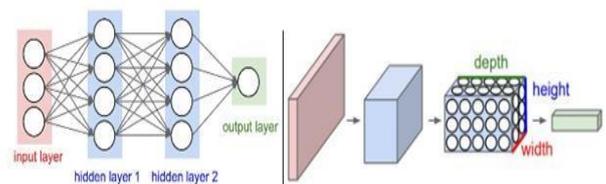
Deep learning memungkinkan model komputasi yang terdiri dari beberapa lapisan pemrosesan untuk mempelajari representasi data dengan berbagai tingkat abstraksi. Metode-metode ini telah secara dramatis meningkatkan kemampuan pengenalan kata-kata, pengenalan objek visual, deteksi objek dan banyak lainnya. *Deep learning* menggali struktur yang rumit

dalam set data yang besar dengan menggunakan algoritma *backpropagation* untuk menunjukkan bagaimana sebuah mesin harus mengubah parameter internal yang digunakan untuk menghitung representasi di setiap lapisan dari representasi dalam lapisan sebelumnya. *Deep learning* mampu melakukan pengenalan pola-pola yang sangat akurat seperti pola grafis, tulisan tangan dan pola lainnya sehingga sangat baik digunakan untuk pengembangan aplikasi-aplikasi seperti pengenalan maupun klasifikasi citra, suara, teks, dan lainnya. Model dalam *Deep learning* biasanya dibangun dengan menggunakan metode Jaringan Syaraf Tiruan (JST) (LeCun dkk., 2015).

3.3 Convolutional Neural Network

Convolutional Neural Network (CNN) adalah pengembangan dari Multilayer Perceptron (MLP) yang didesain untuk mengolah data dua dimensi. CNN termasuk dalam jenis *deep neural network* karena kedalaman jaringan yang tinggi dan banyak diaplikasikan pada data citra. Pada kasus klasifikasi citra, MLP kurang sesuai untuk digunakan karena tidak menyimpan informasi spasial dari data citra dan menganggap setiap piksel adalah fitur yang independen sehingga menghasilkan citra yang kurang baik (Sharif Razavian dkk., 2014).

Konsep CNN pada dasarnya memiliki persamaan dengan cara kerja MLP. Namun dalam CNN, neuron-neuron dipresentasikan ke dalam bentuk tiga dimensi, sedangkan MLP, setiap neuron berukuran satu dimensi. Sehingga, menyebabkan parameter bobot pada CNN dan MLP berbeda. MLP memiliki bobot satu dimensi, yaitu jumlah dari *neuron input*. Operasi *linear* pada CNN menggunakan operasi konvolusi dan bobot bukan berbentuk satu dimensi melainkan berbentuk tiga dimensi, yaitu *pixel* tinggi, *pixel* lebar dan *channel* warna. Gambar 1 menunjukkan arsitektur dari CNN (Shafira, 2018).

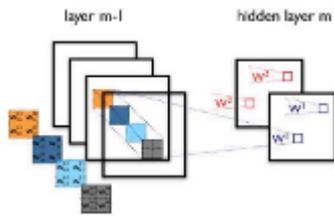


Gambar 1. Arsitektur Convolutional Neural Network

Pada CNN, data yang dipropagasikan pada jaringan adalah data dua dimensi, sehingga operasi linear dan parameter bobot pada CNN berbeda. Pada CNN operasi linear menggunakan operasi konvolusi, sedangkan bobot tidak lagi satu dimensi saja, namun berbentuk empat dimensi yang merupakan kumpulan *kernel* konvolusi seperti pada Gambar.2 Dimensi bobot pada CNN adalah:

$$\text{neuron input} \times \text{neuron output} \times \text{tinggi} \times \text{lebar} \quad (1)$$

Karena sifat proses konvolusi, maka CNN hanya dapat digunakan pada data yang memiliki struktur dua dimensi seperti citra dan suara (Putra, 2016).



Gambar 2. Proses Konvolusi pada CNN

3.4 Arsitektur Jaringan CNN

JST terdiri dari berbagai layer dan beberapa neuron pada masing-masing layer. Kedua hal tersebut tidak dapat ditentukan menggunakan aturan yang pasti dan berlaku berbeda-beda pada data yang berbeda (Aghdam & Heravi, 2017).

Pada kasus MLP, sebuah jaringan tanpa *hidden layer* dapat memetakan persamaan linear apa pun, sedangkan jaringan dengan satu atau dua *hidden layer* dapat memetakan sebagian besar persamaan pada data sederhana. Namun pada data yang lebih kompleks, MLP memiliki keterbatasan. Pada permasalahan jumlah *hidden layer* di bawah tiga *layer*, terdapat pendekatan untuk menentukan jumlah neuron pada masing-masing *layer* untuk mendekati hasil optimal. Penggunaan *layer* di atas dua pada umumnya tidak direkomendasikan dikarenakan akan menyebabkan *overfitting* serta kekuatan *backpropagation* berkurang secara signifikan. Dengan berkembangnya *deep learning*, ditemukan bahwa untuk mengatasi kekurangan MLP dalam menangani data kompleks, diperlukan fungsi untuk mentransformasi data input menjadi bentuk yang lebih mudah dimengerti oleh MLP. Hal tersebut memicu berkembangnya *deep learning* di mana dalam satu model diberi beberapa *layer* untuk melakukan transformasi data sebelum data diolah menggunakan metode klasifikasi. Hal tersebut memicu berkembangnya model *neural network* dengan jumlah layer di atas tiga. Namun dikarenakan fungsi *layer* awal sebagai metode ekstraksi fitur, maka jumlah layer dalam sebuah DNN tidak memiliki aturan universal dan berlaku berbeda beda tergantung *dataset* yang digunakan (Putra, 2016).

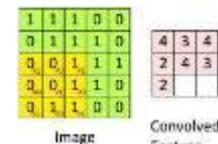
Karena hal tersebut, jumlah layer pada jaringan serta jumlah neuron pada masing-masing *layer* dianggap sebagai *hyperparameter* dan dioptimasi menggunakan pendekatan *searching*. Sebuah CNN terdiri dari beberapa layer. Berdasarkan arsitektur LeNet5 (Stanford University, 2013), terdapat empat macam *layer* utama pada sebuah CNN namun yang diterapkan hanya tiga macam lapisan antara lain:

1. Convolution Layer

Convolution Layer melakukan operasi konvolusi pada *output* dari *layer* sebelumnya. *Layer* tersebut adalah proses utama yang mendasari sebuah CNN.

Konvolusi adalah suatu istilah matematis yang berarti mengaplikasikan sebuah fungsi pada *output* fungsi lain secara berulang. Dalam pengolahan citra, konvolusi berarti mengaplikasikan sebuah *kernel* (kotak kuning) pada citra di semua *offset* yang memungkinkan seperti yang ditunjukkan pada Gambar.3. Kotak hijau secara keseluruhan adalah citra yang akan dikonvolusi. *Kernel* bergerak dari sudut kiri atas ke kanan bawah. Sehingga hasil konvolusi dari citra tersebut dapat dilihat pada gambar di sebelah kanannya.

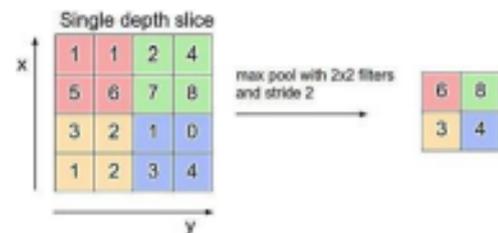
Tujuan dilakukannya konvolusi pada data citra adalah untuk mengekstraksi fitur dari citra *input*. Konvolusi akan menghasilkan transformasi linear dari data *input* sesuai informasi spasial pada data. Bobot pada layer tersebut menyempitkan *kernel* konvolusi yang digunakan, sehingga *kernel* konvolusi dapat dilatih berdasarkan input pada CNN (Putra, 2016).



Gambar 3. Operasi Konvolusi

2. Subsampling Layer

Subsampling adalah proses mereduksi ukuran sebuah data citra. Dalam pengolahan citra, sub sampling juga bertujuan untuk meningkatkan invariansi posisi dari fitur. Dalam sebagian besar CNN, metode sub sampling yang digunakan adalah *max pooling*. *Max pooling* membagi *output* dari *convolution layer* menjadi beberapa *grid* kecil lalu mengambil nilai maksimal dari setiap *grid* untuk menyusun matriks citra yang telah direduksi seperti yang ditunjukkan pada Gambar.4. Grid yang berwarna merah, hijau, kuning dan biru merupakan kelompok *grid* yang akan dipilih nilai maksimumnya. Sehingga hasil dari proses tersebut dapat dilihat pada kumpulan *grid* *disebelah* kanannya. Proses tersebut memastikan fitur yang didapatkan akan sama meskipun objek citra mengalami translasi (pergeseran) (Putra, 2016).



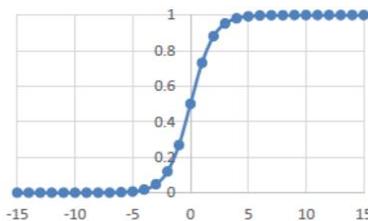
Gambar 4. Operasi Max Pooling

Menurut (Springenberg dkk., 2014), penggunaan *pooling layer* pada CNN hanya bertujuan untuk mereduksi ukuran citra sehingga dapat dengan mudah

digantikan dengan sebuah *convolution layer* dengan *stride* yang sama dengan *pooling layer* yang bersangkutan.

3. *Fully Connected Layer*

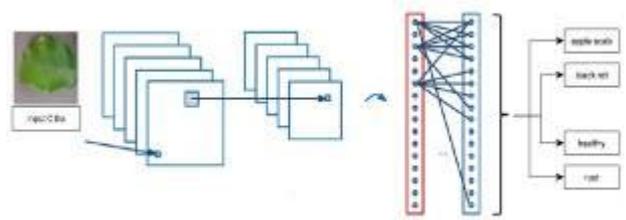
Layer tersebut adalah layer yang biasanya digunakan dalam penerapan MLP dan bertujuan untuk melakukan transformasi pada dimensi data agar data dapat diklasifikasikan secara *linear*. Setiap *neuron* pada *convolution layer* perlu ditransformasi menjadi data satu dimensi terlebih dahulu sebelum dapat dimasukkan ke dalam sebuah *fully connected layer*. Karena hal tersebut menyebabkan data kehilangan informasi spasialnya dan tidak *reversibel*, *fully connected layer* hanya dapat diimplementasikan di akhir jaringan. *Convolution layer* dengan ukuran kernel 1 x 1 melakukan fungsi yang sama dengan sebuah *fully connected layer* namun dengan tetap mempertahankan karakter spasial dari data. Hal tersebut membuat penggunaan *fully connected layer* pada CNN sekarang tidak banyak dipakai. Fungsi sigmoid (gambar 5) sekarang sudah tidak banyak digunakan dalam praktik karena memiliki kelemahan utama yaitu range nilai *output* dari fungsi *sigmoid* tidak terpusat pada angka nol. Hal tersebut menyebabkan terjadinya proses *backpropagation* yang tidak ideal, selain itu bobot pada JST tidak terdistribusi rata antara nilai positif dan negatif serta nilai bobot akan banyak mendekati ekstrim 0 atau 1. Dikarenakan komputasi nilai propagasi menggunakan perkalian, maka nilai ekstrem tersebut akan menyebabkan efek *saturating gradients* di mana jika nilai bobot cukup kecil, maka lama kelamaan nilai bobot akan mendekati salah satu ekstrem sehingga memiliki gradien yang mendekati nol. Jika hal tersebut terjadi, maka *neuron* tersebut tidak akan dapat mengalami *update* yang signifikan dan akan nonaktif.



Gambar 5. Distribusi Fungsi Sigmoid

4. PEMBAHASAN

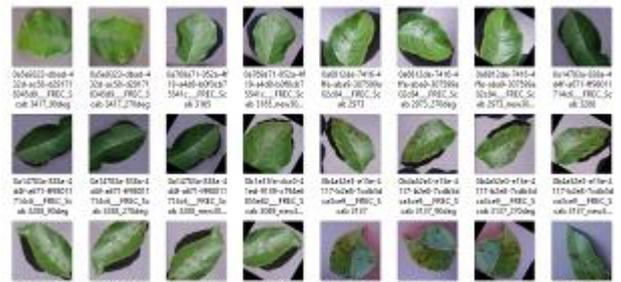
Berikut ini adalah perancangan aplikasi yang menjelaskan proses dari awal aplikasi berjalan hingga aplikasi selesai. Secara umum alur aplikasi yang dirancang pada penelitian ini dapat terlihat pada gambar 6.



Gambar 6. Alur Aplikasi

4.1 Dataset

Pada pengumpulan *dataset* daun Apel, data diambil dari *datasets* website *keggel*, peneliti telah menyusun data ke dalam direktori *train* dan *valid*. Terdapat 7700 data daun Apel dalam folder *train* dan 1943 data daun Apel dalam folder *valid* yang di bagi ke dalam 4 kelas (lihat gambar 7 sampai 10) yaitu *apple scab*, *black rot*, *rust* dan *healthy* (Bisong, 2019)



Gambar 7. Dataset penyakit *apple scab* pada direktori *train*

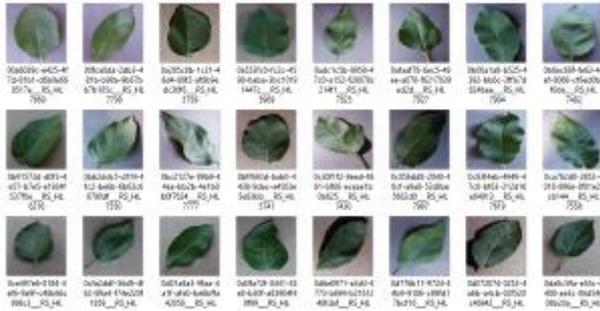


Gambar 8. Dataset penyakit *apple black rot* pada direktori *train*



6.

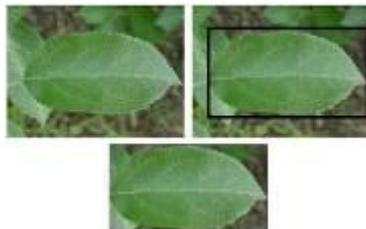
Gambar 9. Dataset penyakit apple rust pada direktori train



Gambar 10. Dataset daun apple healthy pada direktori train

4.2 Pra Proses Input

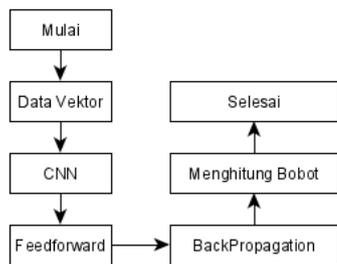
Citra masukan akan diolah ke dalam pra-proses yaitu proses *wrapping* dan *cropping*. Pada *wrapping*, citra masukan dilakukan pengecekan terhadap *edge* dari objek utama pada citra tersebut. Dari *edge* pada citra tersebut ditentukan *edge* maksimalnya sehingga saat hasil *cropping* objek pada citra tersebut tetap utuh seperti pada Gambar 11.



Gambar 11. Crop Citra Input

4.3 Proses Training

Proses *training* merupakan tahapan di mana CNN dilatih untuk memperoleh akurasi yang tinggi dari klasifikasi yang dilakukan. Tahapan ini terdiri dari proses *feed forward* dan proses *backpropagation*.



Gambar 12. Alur Proses Training

4.4 Proses Testing

Proses *testing* merupakan proses klasifikasi menggunakan bobot hasil proses *training*. Proses ini tidak jauh berbeda dengan proses *training* yang membedakannya tidak terdapat proses *backpropagation* setelah proses *feedforward*. Sehingga hasil akhir dari proses ini menghasilkan akurasi dari klasifikasi yang dilakukan, data yang gagal diklasifikasi, nomor citra

yang gagal diklasifikasi, dan bentuk *network* yang terbentuk dari proses *feedforward*. Dengan bobot yang baru proses *feedforward* diterapkan yang kemudian menghasilkan lapisan *output*. Lapisan output sudah *fully connected* dengan label yang disediakan. Hasil *fully connected* tersebut diperoleh data yang gagal dan berhasil diklasifikasi

4.5 Implementasi dan Uji Coba

Pada tahap ini, dilakukan proses klasifikasi terhadap objek untuk mengetahui hasil deteksi yang ditampilkan oleh aplikasi.

Apabila yang dihasilkan oleh aplikasi sudah sesuai dengan yang diharapkan, maka model yang digunakan pada aplikasi sudah benar. Akan tetapi, apabila yang dihasilkan tidak sesuai dengan harapan, maka dapat dipastikan bahwa masih terdapat kesalahan pada model yang dilatih. Hasil dari uji coba akurasi disajikan di dalam bentuk tabel yang dapat dilihat pada Tabel 2.

Tabel 2. Hasil Uji Coba

Objek Asli	Prediksi dan Akurasi Model	Kesimpulan
Apple Scab	 Apple Scab pilih gambar dari gallery ambil gambar menggunakan kamera	Akurat
Apple black rot	 Apple black rot pilih gambar dari gallery ambil gambar menggunakan kamera	Akurat
Apple cedar rust	 Apple cedar rust pilih gambar dari gallery ambil gambar menggunakan kamera	Akurat
Apple healthy	 Healthy Apple pilih gambar dari gallery ambil gambar menggunakan kamera	Akurat

Berdasarkan hasil uji coba yang telah dilakukan, dapat dilihat bahwa aplikasi yang dibuat dapat berfungsi sesuai dengan yang diharapkan sehingga dapat disimpulkan bahwa aplikasi ini telah bekerja dengan benar.

5. KESIMPULAN

Aplikasi ini sudah berhasil dibuat menggunakan *Flutter* dan *Visual Studio Code* dan dapat dijalankan pada *smartphone* dengan minimum versi *Android Lollipop*. Data Pelatihan citra dengan algoritma *convolutional neural network* yang berasal dari 7700 data train dan 1943 data validasi. Proses *preprocessing* yang dilakukan pada citra adalah dengan melakukan perubahan ukuran citra agar berukuran sama yaitu menjadi 256x256 dan menggunakan tipe gambar RGB. Model dapat mengklasifikasi citra yang diujikan dengan baik, yaitu penyakit *apple scab*, *apple black rot*, *apple cedar rust* dan tanaman apel sehat. Hasil dari final *test accuracy* yang diperoleh dari keseluruhan proses pelatihan yaitu sebesar 97,1%.

6. SARAN

Pengembangan lebih lanjut dapat dilakukan untuk menyempurnakan aplikasi untuk mengenali jenis penyakit tanaman Apel yaitu dengan menambahkan jumlah jenis penyakit serta jumlah gambar pada *dataset* untuk memperoleh hasil akurasi yang lebih maksimal dan memberikan penjelasan serta penanganan yang benar untuk penyakit tertentu.

7. DAFTAR PUSTAKA

- Bisong, E. (2019). *Building Machine Learning and Deep learning Models on Google Cloud Platform*. Springer.
- Ernawati. (2018). Karakteristik Uji Organoleptik Pembuatan Cup Cake Menggunakan Tepung Beras Merah. *Jurnal Hospitality. Jurnal Hospitality, ISSN:2338-7440*.
- LeCun, Y., Bengio, Y., & Hinton, G. (2015). *Deep learning. Nature, 521(7553), 436–444*.
- Permana, A. H., Asmara, R. A., & Tri, A. R. (2015). Sistem Pakar Diagnosa Hama Dan Penyakit Pada Tanaman Apel Menggunakan Metode Certainty Factor. *Jurnal Informatika Polinema, 1(3), 7*.
- Putra, I. (2016). *Klasifikasi citra menggunakan convolutional neural network (CNN) pada caltech 101*. Institut Teknologi Sepuluh Nopember.
- Shafira, T. (2018). *Implementasi Convolutional Neural Networks Untuk Klasifikasi Citra Tomat Menggunakan Keras*. Universitas Islam Indonesia.
- Sharif Razavian, A., Azizpour, H., Sullivan, J., & Carlsson, S. (2014). CNN features off-the-shelf: an astounding baseline for recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 806–813*.
- Springenberg, J. T., Dosovitskiy, A., Brox, T., & Riedmiller, M. (2014). Striving for simplicity: The all convolutional net. *ArXiv Preprint ArXiv:1412.6806*.
- Stanford University. (2013). *An Introduction to Convolutional Neural Networks*. Stanford University.
http://scarlet.stanford.edu/teach/index.php/An_Introduction_to_Convolutional_Neural_Networks
- Aghdam, H. H., & Heravi, E. J. (2017). *Guide to convolutional neural networks*. New York, NY: Springer, 10, 978-973.
- Suhardjo, H. L., Deaton, B. J., & Driskel, J. A. (1985). *Pangan, Gizi dan Pertanian. UI-Pers Jakarta*.