

# Penerapan Metode *Convolutional Neural Network* untuk Klasifikasi Penyakit Daun Apel pada *Imbalanced Data*

Chilyatun Nisa<sup>1</sup>, Eva Yulia Puspaningrum<sup>2\*</sup>, Hendra Maulana<sup>3</sup>

<sup>1,3</sup>Informatika, Universitas Pembangunan Nasional “Veteran” Jawa Timur

<sup>1</sup>[17081010050@student.upnjatim.ac.id](mailto:17081010050@student.upnjatim.ac.id)

<sup>3</sup>[hendra.maulana.if@upnjatim.ac.id](mailto:hendra.maulana.if@upnjatim.ac.id)

<sup>2</sup>Informatika, Universitas Pembangunan Nasional “Veteran” Jawa Timur

\*Corresponding author email: [evapuspaningrum@upnjatim.ac.id](mailto:evapuspaningrum@upnjatim.ac.id)

**Abstrak**— Menurut data produksi buah-buahan di Indonesia, produksi apel pada tahun 2017 mengalami penurunan sebesar 3,3% atau sejumlah 10.780 ton dari tahun 2016 yang menghasilkan sebanyak 329.780 ton. Hal itu disebabkan oleh berbagai penyakit yang sering terjadi pada produksi apel, oleh karena itu pendeteksian penyakit daun apel yang tepat waktu menjadi sangat penting untuk industri apel yang berkembang dengan sehat. Sehingga dibutuhkan sistem yang efektif seperti klasifikasi citra digital pada tanaman. Metode yang digunakan pada penelitian ini merupakan adalah *Convolutional Neural Network (CNN)* dengan arsitektur *InceptionV3*. Penelitian ini menggunakan *dataset Plant Pathology 2020 - FGV C7* sebanyak 1.821 data citra dengan 4 kelas. Data dibagi menjadi 3 set data (latih, validasi, dan uji) dengan rasio 70:10:20. Hasil performa dievaluasi dengan data uji menggunakan *confusion matrix*. Dari hasil pelatihan didapatkan akurasi 96,37%. Pada pengujian hasil akurasi pada masing-masing kelas sebesar 90,6%, 62,3%, 94,3%, dan 92%.

**Kata Kunci**— Penyakit Daun Apel, Klasifikasi Citra Digital, *Convolutional Neural Network*, *InceptionV3*, *Confusion Matrix*.

## I. PENDAHULUAN

Tanaman apel atau yang memiliki nama latin *Malus sylvestris*, merupakan tanaman tahunan dari daerah subtropis yang dapat dibudidayakan di Indonesia [1]. Dengan nilai gizi dan obat yang tinggi, apel termasuk salah satu jenis buah yang paling produktif di dunia [2].

Menurut data produksi buah-buahan di Indonesia pada tahun 2013-2017 yang diterbitkan oleh Kementerian Pertanian Republik Indonesia, dengan total produksi tanaman apel pada tahun 2017 sebanyak 319.000 ton. Namun, produksi apel pada tahun 2017 mengalami penurunan sebesar 3,3% atau sejumlah 10.780 ton dari tahun 2016 yang menghasilkan sebanyak 329.780 ton [3]. Hal itu disebabkan oleh berbagai penyakit yang sering terjadi pada produksi apel. Sehingga deteksi penyakit daun apel yang efektif dan efisien sangat diperlukan untuk memastikan perkembangan apel yang sehat [2].

Secara tradisional, observasi visual oleh para ahli telah dilakukan untuk mendiagnosis penyakit tanaman. Namun, ada resiko kesalahan karena persepsi subjektif [4]. Dalam konteks

ini, berbagai teknik spektroskopi dan pencitraan telah dipelajari untuk mendeteksi penyakit tanaman. Tetapi, mereka membutuhkan instrumen yang tepat dan sensor yang besar yang menyebabkan biaya tinggi dan efisiensi rendah [5]. Dalam beberapa tahun terakhir, dengan populernya kamera digital dan perangkat elektronik lain, diagnosis penyakit tanaman otomatis melalui pembelajaran mesin telah banyak diterapkan sebagai alternatif yang memuaskan [6].

Dengan berbagai alasan yang telah dipaparkan, penyakit tanaman harus dideteksi untuk mencegah penyebarannya ke pohon lain. Deteksi penyakit dan hama tanaman berdasarkan pemrosesan gambar dan visi komputer menjadi semakin penting, dengan berbagai penelitian yang dilakukan selama 15 tahun terakhir tentang deteksi penyakit dan hama tanaman [7].

Berikut beberapa penelitian terdahulu, seperti yang dilakukan oleh Nachtigall, Lucas G., et al tahun 2016 [8] menggunakan algoritma CNN dengan arsitektur AlexNet untuk mendeteksi dan mengklasifikasikan kerusakan herbisida pada daun apel. *Dataset* diperoleh dengan memotret setiap daun di atas latar belakang putih yang terdiri dari 2539 gambar dari 6 kelas, menghasilkan akurasi sebesar 97%.

Penerapan algoritma CNN yang dilakukan oleh Fang Tao., et al tahun 2019 [9] yang menggabungkan *batch normalization* dan fungsi *central loss* berdasarkan arsitektur VGG16 untuk klasifikasi penyakit daun apel. Data yang digunakan berupa citra daun yang sakit sebanyak 5.373 citra dan citra daun sehat sebanyak 1.683. Model yang diusulkan mendapatkan akurasi pengujian sebesar 95,0%.

Penerapan algoritma CNN juga dilakukan oleh Baranwal Saraansh., et al pada tahun 2019 [10] dengan menggunakan arsitektur GoogleNet untuk mendeteksi penyakit daun apel. Menggunakan *dataset PlantVillage* yang terdiri dari empat kelas, tiga diantaranya daun berpenyakit sejumlah 1.526 dan daun sehat sejumlah 1.000 sampel. Model mendapat akurasi sebesar 98,54%.

Penelitian serupa juga dilakukan oleh Wicaksono Guntur., et al tahun 2020 [11] menggunakan algoritma CNN dengan arsitektur LeNet-5 untuk pengidentifikasian penyakit tanaman apel. Penelitian ini menggunakan *dataset PlantVillage* sejumlah 3.151 citra daun yang telah diklasifikasikan. Mendapatkan hasil pengujian, akurasi model sebesar 99,4% dan akurasi validasi 97,8% pada *epoch* 100.

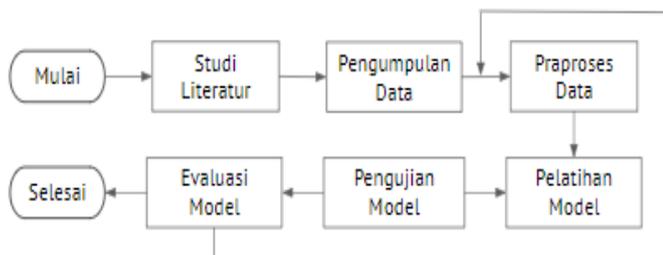
Penelitian lain dilakukan oleh Zhong Yong., et al pada tahun 2020 [12]. Dalam penelitian ini, menggunakan CNN dengan DenseNet-121 dan tiga metode regresi, klasifikasi

multi-label, dan fungsi loss untuk identifikasi penyakit daun apel. *Dataset* yang digunakan berasal dari *AIChallenger-Plant Disease Recognition*. Kumpulan data citra daun apel, termasuk 2.462 citra enam kelas penyakit daun apel. Metode yang diusulkan mencapai akurasi sebanyak 93,51%, 93,31%, dan 93,71% pada set pengujian.

Pada penelitian ini penulis mengusulkan model dengan menggunakan arsitektur *InceptionV3*. Selain itu, pada penelitian terdahulu banyak yang menggunakan *dataset PlantVillage* dimana data tersebut sudah dirilis sejak tahun 2018 yang lalu. Sehingga pada penelitian ini penulis menggunakan *dataset Plant Pathology 2020 - FGVC7* yang dirilis pada tahun 2020, yang mana belum ada penelitian yang menggunakan *dataset* ini kecuali penelitian terkait *dataset* itu sendiri.

## II. METODOLOGI PENELITIAN

Metodologi penelitian yang dilakukan sebagaimana ditunjukkan pada Gbr. 1, dimulai dengan melakukan studi literatur kemudian melakukan pengumpulan data. Dilanjutkan melakukan praproses data, kemudian pelatihan model pada data latih dan pengujian model pada data uji. Setelah itu barulah dilakukan evaluasi model pada tahap terakhir. Jika hasil evaluasi tidak bagus, maka proses akan diulang lagi ke tahap praproses data untuk penyetelan *hyperparameter* hingga didapatkan hasil evaluasi yang bagus.



Gbr. 1 Metode penelitian

### A. Dataset

Penelitian ini menggunakan *dataset Plant Pathology 2020 - FGV C7* [13], memiliki jumlah data sebanyak 1.821 citra daun apel yang diklasifikasikan berdasarkan kelas masing-masing. Data ini merupakan data citra berwarna yang berekstensi .jpg pada setiap gambarnya.

Kelas data terdiri dari empat kelas yaitu satu kelas daun sehat dan tiga kelas daun berpenyakit (*healthy*, *multiple disease*, *rust*, dan *scab*). Pada Tabel 1 dijelaskan mengenai deskripsi kelas daun apel yang sehat dan berpenyakit yang akan diteliti dengan contoh citra daun apel yang telah dideteksi penyakitnya disertai dengan penyebab dan gejala yang timbul pada citra daun apel berdasarkan pada penelitian yang telah dilakukan sebelumnya [11].

Selanjutnya *dataset* tersebut akan dibagi menjadi tiga set yaitu data latih, data validasi, dan data uji dengan rasio 70:10:20 pada keseluruhan data.

TABEL 1  
DESKRIPSI *DATASET*

| Jenis                   | Citra  | Deskripsi   |
|-------------------------|--|---|
| <i>Apple Rust</i>       |    | Penyakit disebabkan oleh jamur <i>Gymnosporangium juniperi-virginiana</i> . Gejala yang muncul pada daun berupa bercak-bercak kuning kecoklatan.  |
| <i>Apple Scab</i>       |    | Penyakit disebabkan oleh jamur <i>Venturia inaequalis</i> . Gejala yang muncul yaitu bercak-bercak hijau yang muncul pada dua sisi daun. Setelah penyakit semakin berkembang, daun akan berwarna hitam/keunguan.  |
| <i>Multiple Disease</i> |   | Penyakit seperti <i>Apple black rot</i> dan <i>Frogeye leaf spot</i> disebabkan oleh jamur <i>Botryosphaeria obtusa</i> . Gejala yang ditimbulkan pada daun berupa bercak-bercak ungu, kekuningan, atau coklat dengan diameter ¼ sampai ½ inci di permukaan daun. |
| <i>Apple Healthy</i>    |  | Daun yang sehat memiliki bentuk oval atau bulat memanjang dengan bagian ujung lancip dan bagian pangkal daun tumpul. Memiliki warna hijau muda atau hijau tua yang tampak segar.  |

*Dataset* yang telah dibagi berdasarkan rasio yang telah disebutkan, akan menghasilkan persebaran data seperti pada Tabel 2.

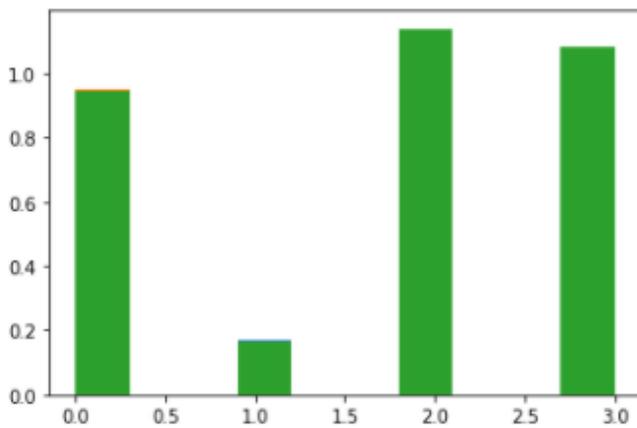
TABEL 2  
*DATASET* DAUN APEL

| Label    | <i>Health</i> | MD | <i>Rust</i> | <i>Scab</i> | Total |
|----------|---------------|----|-------------|-------------|-------|
| Latih    | 361           | 64 | 436         | 415         | 1276  |
| Validasi | 52            | 9  | 62          | 59          | 182   |
| Uji      | 103           | 18 | 124         | 118         | 363   |

Berdasarkan uraian tersebut, dapat terlihat bahwa data yang penulis gunakan merupakan data yang tidak seimbang. Karena jumlah data tiap kelas pada set data berbeda-beda.

### B. Praproses Data

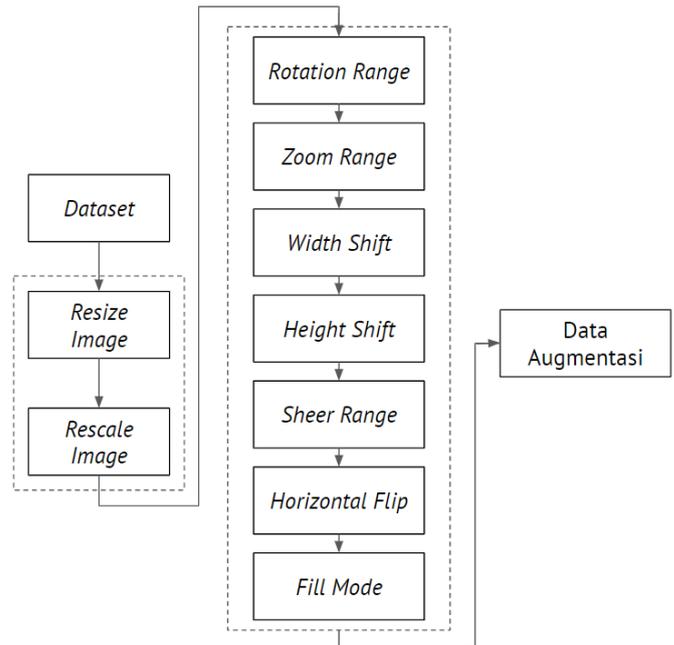
Pada tahap ini akan dilakukan praproses data berupa merubah ukuran citra menjadi  $256 \times 256$  piksel yang memiliki tiga *channel* yakni *Red*, *Green*, and *Blue* (RGB). Sehingga dihasilkan sebuah bentuk dari setiap citra pada *height*, *weight*, dan *channel* menjadi  $256 \times 256 \times 3$ . Selanjutnya juga dilakukan penskalaan ulang pada input yang berupa koefisien berwarna yang bernilai antara 0-255 dengan perhitungan  $\text{input} \times 1/255$  untuk mengubah nilai menjadi antara 0 sampai 1 [14]. Data yang penulis gunakan pada penelitian ini merupakan data yang tidak seimbang (*imbalanced data*), dapat dilihat dari persebaran kelas yang ada pada Gbr. 2. Terlihat jelas bahwa persebaran data pada setiap kelasnya tidak merata. Sehingga perlu dilakukan *oversampling* untuk menghindari terjadinya *overfitting* pada model.



Gbr. 2 Grafik Imbalanced Data

Untuk melakukan *oversampling* sekaligus menghindari adanya *overfitting* maka penulis melakukan teknik augmentasi pada data. Adapun berikut merupakan beberapa metode yang dilakukan penulis pada dataset terkait pada penelitian sebagaimana yang ada pada Gbr. 3. Seperti ada *rotation range*, *zoom range*, *width shift*, *sheer range*, *horizontal flip*, dan *fill mode* yang dijelaskan pada penjelasan di bawah ini :

1. *Rotation range*, yaitu dengan merotasi atau melakukan perputaran pada gambar.
2. *Zoom range*, yaitu dengan melakukan perbesaran pada gambar.
3. *Width shift range*, yaitu dengan melakukan pergeseran lebar pada gambar.
4. *Shear range*, yaitu dengan melakukan pergeseran tinggi pada gambar.
5. *Horizontal flip*, yaitu dengan membalik gambar secara horizontal
6. *Fill mode*, yaitu dengan mengisi ruang kosong pada gambar.



Gbr. 3 Augmentasi Data

### C. Convolutional Neural Network (CNN)

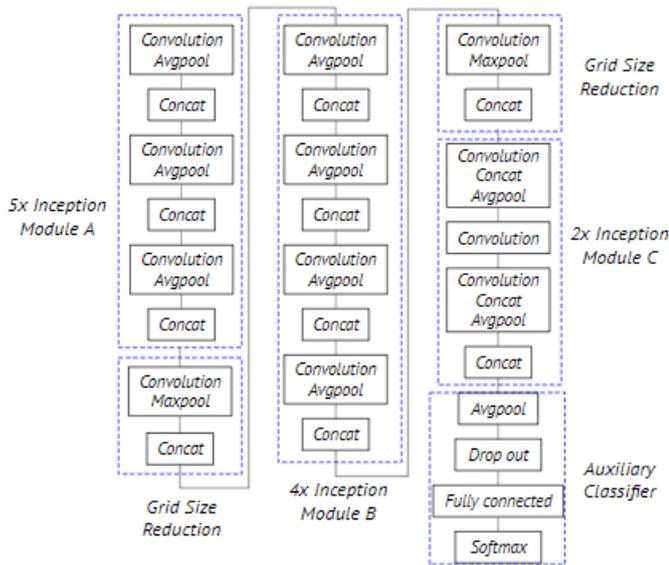
Algoritma *Convolutional Neural Network* atau CNN adalah salah satu algoritma populer yang dipakai untuk pembelajaran mendalam [15][16]. CNN juga dapat digunakan untuk pelatihan data, baik secara *supervised learning* maupun *unsupervised learning* [17]. Algoritma ini juga biasa digunakan untuk melakukan klasifikasi citra [15].

Pada penelitian ini penulis menggunakan arsitektur *InceptionV3* pada pengerjaannya. *InceptionV3* adalah sebuah model *deep convolutional neural network* yang dikembangkan oleh Google untuk memenuhi *ImageNet Large Visual Recognition Challenge* pada tahun 2012. *Inception* sendiri memiliki empat versi, yaitu *InceptionV1*, *InceptionV2*, *InceptionV3*, dan *InceptionV4*.

Model *Inception* menggunakan beberapa *filter* pada *layer* yang biasa. Hasil dari beberapa *filter* tersebut dijadikan satu lagi menggunakan *channel concat* sebelum masuk ke dalam iterasi berikutnya [18].

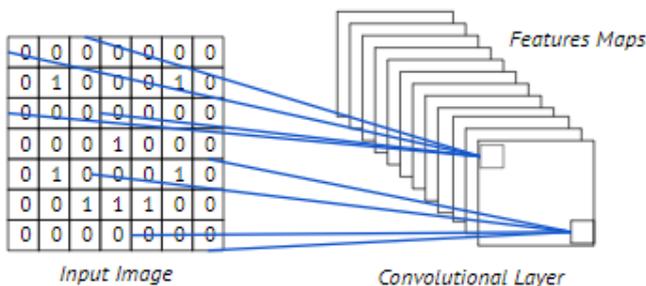
Jaringan *InceptionV3* memiliki beberapa blok bangunan simetris dan asimetris, dimana setiap blok memiliki beberapa cabang konvolusi, penyatuan rata-rata, *max pooling*, digabungkan, putus, dan lapisan yang terhubung sepenuhnya [19]. Arsitekturnya diilustrasikan seperti pada Gbr. 4.

*InceptionV3* adalah arsitektur pertama dengan parameter yang lebih sedikit dan komputasi yang efisien [20]. *InceptionV3* merupakan gabungan dari perbaikan versi pertama dan kedua, dimana terdapat beberapa tambahan seperti *RMSProp optimizer*, *factorized 7x7 convolutions*, *batch normal* pada *auxiliary classifier*, dan *label smoothing* [21]. Pada *InceptionV3* ini ada faktorisasi untuk mengurangi parameter. Meskipun komputasinya rendah, *InceptionV3* tidak dapat digunakan pada perangkat dengan ruang lingkup komputasi yang rendah.



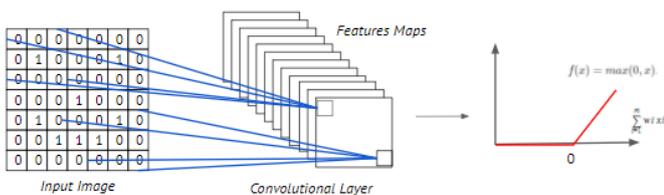
Gbr. 4 Arsitektur InceptionV3

Lapisan konvolusional seperti yang diilustrasikan pada Gbr. 5. yang merupakan bagian utama dari arsitektur CNN. Pada tahap ini akan dilakukan operasi konvolusi pada keluaran fungsi lain secara berulang. Adapun tujuan dari konvolusi adalah untuk mengekstrak fitur dari data citra masukan. Selanjutnya konvolusi akan menghasilkan sebuah transformasi linier dari data masukan. Kemudian bobot pada lapisan tersebut juga menentukan kernel konvolusi mana yang digunakan, sehingga kernel konvolusi dapat dilatih berdasarkan pada masukan di CNN [22].



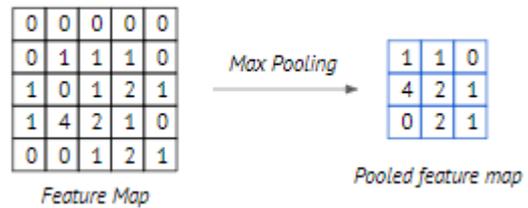
Gbr. 5 Lapisan Convolutional

Pada Gbr. 6 diilustrasikan terdapat fungsi aktivasi yang akan digunakan, yaitu Rectified Linear Unit atau ReLU. Kernel atau filter yang digunakan untuk setiap lapisan konvolusi berukuran  $3 \times 3$  dengan tujuan agar mempercepat proses pelatihan dan meningkatkan hasil akurasi [22].



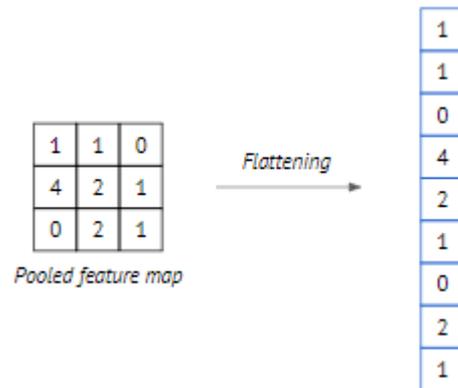
Gbr. 6 Fungsi Aktivasi ReLU

Pooling pada Gbr. 7 merupakan proses yang mereduksi ukuran spasial dari hasil pada lapisan konvolusi yang digunakan sebagai input untuk menghasilkan representasi fitur. Hal ini dapat memberi CNN kemampuan untuk mengenali objek. Lapisan pooling dapat digunakan untuk mengambil nilai maksimum dengan max-pooling atau nilai rata-rata bagian piksel pada gambar [22].



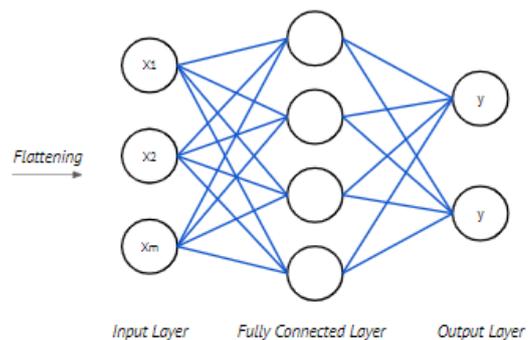
Gbr. 7 Lapisan Max Pooling

Setelah tahap pada lapisan pooling selesai, maka akan dilakukan tahap selanjutnya yaitu perataan atau level hasil dari lapisan pooling ke dalam lapisan yang terhubung sepenuhnya. Seperti yang telah diilustrasikan pada Gbr. 8.



Gbr. 8 Lapisan Flattening

Lapisan Fully Connected pada Gbr. 9 adalah lapisan yang ketika semua neuron aktivasi dari lapisan sebelumnya terhubung dengan neuron di lapisan berikutnya secara keseluruhan. Aktivasi dari lapisan sebelumnya harus diubah menjadi data satu dimensi sebelum dapat dihubungkan dengan semua neuron di lapisan Fully Connected [22].



Gbr. 9 Lapisan Fully Connected

#### D. Latih Model

Proses pelatihan data dapat dilihat pada Gbr. 10, yang dimulai dari data augmentasi, kemudian menentukan arsitektur algoritma beserta penyetelan parameter. Pada penelitian ini parameter yang digunakan seperti  $epoch = 14$ ,  $optimizer$  menggunakan Adam dengan  $learning rate$  sebesar  $0.0001$ ,  $batch size = 64$ ,  $loss function = categorical cross entropy$ , dan lain-lain. Setelah itu lanjut proses pelatihan data yang dilakukan secara berulang sesuai jumlah  $epoch$  yang telah ditentukan, sesuai dengan kebutuhan.

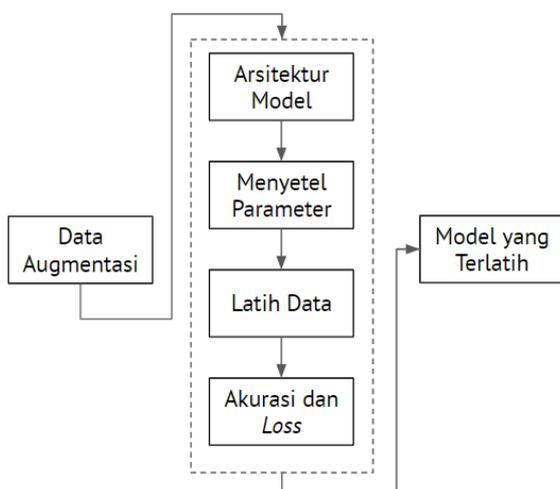
Adam optimizer merupakan algoritma optimalisasi yang berasal dari RMSProp dan *Stochastic Gradient Descent* (SGD) [23]. Metode ini banyak digunakan karena mudah untuk diterapkan, efisien secara komputasi sehingga tidak membutuhkan banyak memori, dan cocok untuk data yang besar [24].

Sedangkan fungsi  $loss$  yang digunakan *categorical cross entropy* merupakan fungsi kerugian yang digunakan dalam tugas klasifikasi *multi-class*. Ini adalah tugas di mana sebuah contoh hanya dapat dimiliki oleh salah satu dari banyak kategori yang memungkinkan, dan model harus memutuskan yang mana. Secara formal, ini dirancang untuk mengukur perbedaan antara dua distribusi probabilitas. Fungsi *loss categorical cross entropy* menghitung kerugian contoh dengan menghitung jumlah pada persamaan (1) berikut [25] :

$$Loss = -\sum_{i=1}^{output\ size} y_i \cdot \log \gamma_i \quad (1)$$

Dimana  $\gamma_i$  dengan  $i$  merupakan nilai skalar dari keluaran model,  $y_i$  merupakan nilai target yang sesuai, dan ukuran keluaran adalah jumlah nilai skalar dalam keluaran model.

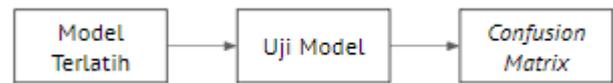
Fungsi  $loss$  ini memiliki ukuran yang sangat baik tentang seberapa dapat dibedakan dua distribusi probabilitas terpisah satu sama lain. Dalam konteks ini  $\gamma_i$  adalah probabilitas dari kejadian  $i$  dan jumlah semuanya dari  $y_i = 1$ , yang artinya tepat pada satu peristiwa dapat terjadi. Kemudian tanda minus memastikan bahwa kerugian semakin kecil.



Gbr. 10 Latih Model

#### E. Evaluasi Model

Proses evaluasi model seperti pada Gbr. 11, dimana menggunakan *confusion matrix* untuk mengetahui persebaran hasil prediksi dibandingkan dengan label yang sebenarnya dari 4 kelas mulai kelas apel *healthy* sampai *scab*.



Gbr. 11 Evaluasi Model

*Confusion matrix* digunakan untuk mengetahui hasil performa dari model yang telah dibuat yaitu dengan *accuracy*, *precision*, *recall*, dan *f1 score* [26] :

- *True Positive* (TP), terjadi jika penyakit daun diprediksi sesuai kelasnya (*Positive*), dan memang sesuai dengan kelasnya (*True*).
- *True Negative* (TN), terjadi jika penyakit daun diprediksi tidak sesuai kelasnya (*Negative*), dan memang tidak sesuai dengan kelasnya (*True*).
- *False Positive* (FP), terjadi jika penyakit daun diprediksi sesuai kelasnya (*Positive*), dan ternyata tidak sesuai dengan kelasnya (*False*).
- *False Negative* (FN), terjadi jika penyakit daun diprediksi tidak sesuai kelasnya (*Negative*), dan memang tidak sesuai dengan kelasnya (*False*).

*Accuracy* adalah rasio dari prediksi yang benar dari keseluruhan data. Akurasi menjelaskan berapa persentase penyakit daun apel yang diprediksi benar sesuai kelasnya. Sebagaimana pada persamaan (2) berikut [26] :

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad (2)$$

*Precision* adalah rasio prediksi yang benar dari keseluruhan hasil yang diprediksi positif. *Precision* menjelaskan berapa persentase penyakit daun apel yang sesuai kelasnya dari seluruh daun yang diprediksi sesuai kelasnya. Sebagaimana pada persamaan (3) berikut [26] :

$$Precision = \frac{TP}{TP + FP} \quad (3)$$

*Recall* adalah rasio dari prediksi yang benar dari keseluruhan data yang memang benar. *Recall* menjelaskan berapa persentase penyakit daun apel yang diprediksi sesuai kelasnya dari keseluruhan daun apel yang sesuai kelasnya. Sebagaimana pada persamaan (4) berikut [26] :

$$Recall = \frac{TP}{TP + FN} \quad (4)$$

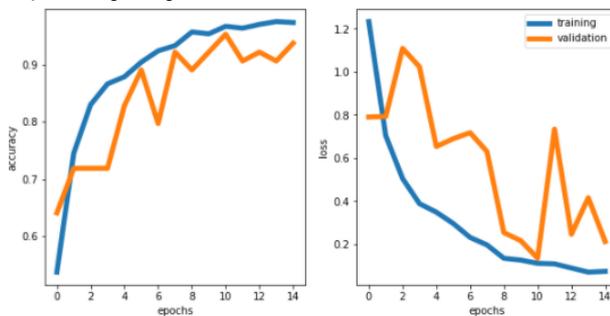
*F1 score* adalah rasio yang membandingkan antara rata-rata *precision* dan *recall*. Sebagaimana pada persamaan (5) berikut [26] :

$$F1\ Score = \frac{2 \times (Recall \times Precision)}{(Recall + Precision)} \quad (5)$$

### III. HASIL DAN PEMBAHASAN

Pengujian yang dilakukan menggunakan perangkat keras dengan CPU Intel Core i7-8565U @1.80GHz~2.GHz, GPU NVIDIA GeForce MX250 4GB, RAM DDR4 8GB, Kemudian untuk perangkat lunak menggunakan sistem operasi Windows 10 64-bit, Python versi 3.8.2, Jupyter Notebook versi 6.0.3, dan Google Colab. Selanjutnya untuk *framework* pembelajaran mendalam menggunakan Keras. Berikut beberapa hasil dari penelitian yang telah dilakukan.

[272] <matplotlib.legend.Legend at 0x7f4bc61e2278>



Gbr. 12 Grafik Akurasi dan Loss

Pada Gbr. 12 merupakan grafik hasil akurasi dan *loss* pada saat pelatihan model yang menggunakan arsitektur *InceptionV3*. Hasilnya cukup baik walaupun terjadi beberapa kali tanjakan atau landaian yang kurang selaras antara akurasi dan validasi.

Akurasi pelatihan dan akurasi validasi tertinggi terjadi pada *epoch* ke-14 yang mencapai nilai 96.37% dan 89.06%. Kemudian untuk *loss* pelatihan dan *loss* validasi mencapai nilai 0.12 dan 0.38.

|                   | precision | recall | f1-score | support |
|-------------------|-----------|--------|----------|---------|
| healthy           | 0.90      | 0.91   | 0.91     | 103     |
| multiple_diseases | 0.60      | 0.17   | 0.26     | 18      |
| rust              | 0.92      | 0.97   | 0.94     | 124     |
| scab              | 0.90      | 0.94   | 0.92     | 118     |
| micro avg         | 0.90      | 0.90   | 0.90     | 363     |
| macro avg         | 0.66      | 0.60   | 0.61     | 363     |
| weighted avg      | 0.89      | 0.90   | 0.89     | 363     |

Gbr. 13 Evaluasi

Hasil dari evaluasi yang menggunakan *confusion matrix* sebagaimana yang ada pada Gbr. 13. Akurasi yang didapatkan berada di angka sekitar 90% pada data uji. Hanya saja terdapat hasil yang cukup signifikan berbeda pada kelas *multiple disease*, hal tersebut bisa terjadi dikarenakan jumlah data kelas tersebut yang memang sedikit sebelum dilakukan augmentasi sehingga data kurang variatif dan penanganan *imbalanced data* pada kelas tersebut kurang baik.

### IV. KESIMPULAN

Dalam makalah ini, dilakukan klasifikasi penyakit pada citra daun apel. Arsitektur CNN yang digunakan merupakan arsitektur *InceptionV3* yang sebelumnya telah dibahas pada bagian implementasi. Sebagian kecil dari pengaturan Model CNN dengan melakukan penyetelan pada *hyperparameter*

seperti nilai *dropout*, ukuran *batch*, dan rasio set data. Akurasi pelatihan terbaik yang dicapai oleh model adalah 96.37% dari keseluruhan *dataset* pada *dropout* 0.5. Akurasi yang sesuai dari kelas apel *healthy*, *multiple disease*, *rust*, dan *scab* masing-masing 90,6%, 62,3%, 94,3%, 92%. Ditemukan bahwa nilai pada kelas kedua tidak cukup bagus, dilihat dari perbedaan nilai yang cukup signifikan daripada ketiga nilai lainnya. Hal ini menunjukkan bahwa penanganan model pada data yang tidak seimbang belum cukup bagus pada salah satu kelas. Untuk penelitian selanjutnya, *dataset* yang digunakan merupakan kelemahan terbesar pada penelitian ini. Karena data tersebut adalah data yang tidak seimbang dan penanganan yang dilakukan hanya dengan augmentasi data, sehingga dibutuhkan penggunaan teknik *oversampling* lain jika ingin menggunakan *dataset* ini. Bisa juga menggunakan *dataset* lain yang seimbang dengan jumlah yang lebih besar, juga bisa mengubah arsitektur maupun *hyperparameter* yang digunakan agar mendapatkan hasil akurasi lebih baik.

### UCAPAN TERIMA KASIH

Penulis sampaikan ucapan terima kasih untuk pihak-pihak yang berkontribusi serta memberikan dukungan. Sehingga penulis dapat menyelesaikan makalah ini dengan tepat waktu.

### REFERENSI

- [1] Anggara, Dewi Shinta Tia, et al. "KENDALA PRODUKSI APEL (Malus sylvestris Mill) Var. MANALAGI DI DESA PONCOKUSUMO KABUPATEN MALANG." *Jurnal Produksi Tanaman*, vol. Vol. 5 No. 2, 2017, pp. 198-207. ISSN: 2527-8452.
- [2] Jiang, Peng, et al. "Real-Time Detection of Apple Leaf Diseases Using Deep Learning Approach Based on Improved Convolutional Neural Networks." *IEEE Access*, vol. Volume 7, no. 2019, 2019. 10.1109/ACCESS.2019.2914929.
- [3] Subdirektorat Statistik Hortikultura. *Statistik Tanaman Buah-Buahan dan Sayuran Tahunan Indonesia 2018*. vol. 5205010, Jakarta, Badan Pusat Statistik, 2019. 05120.1901
- [4] Dutot, M., et al. "Predicting the spread of postharvest disease in stored fruit, with application to apples." *Postharvest biology and technology*, vol. 85, no. 4, 2013.
- [5] Yuan, Lin, et al. "Spectral analysis of winter wheat leaves for detection and differentiation of diseases and insects." *Field Crops Research*, vol. 156, 2014, pp. 199-207. <https://doi.org/10.1016/j.fcr.2013.11.012>.
- [6] Qin, Feng, et al. "Identification of Alfalfa Leaf Diseases Using Image Recognition Technology." 2016. <https://doi.org/10.1371/journal.pone.0168274>.
- [7] Torkoglu, Muammer, et al. "Multi-model LSTM-based convolutional neural networks for detection of apple diseases and pests." *Ambient Intelligent and Humanized Computing*, 2019. [10.1007/s12652-019-01591-w](https://doi.org/10.1007/s12652-019-01591-w).
- [8] Nachtigall, Lucas G., et al. "Classification of Apple Tree Disorders Using Convolutional Neural Networks." *IEEE Computer Society, no. IEEE 28th International on Tools with Artificial Intelligence*, 2016. DOI 10.1109/ICTAI.2016.75.
- [9] Fang, Tao, et al. "Identification of Apple Leaf Diseases Based on Convolutional Neural Network." *Springer Nature Switzerland*, no. ICIC 2019, LNCS, 2019, pp. 553-564. [https://doi.org/10.1007/978-3-030-26763-6\\_53](https://doi.org/10.1007/978-3-030-26763-6_53).
- [10] Baranwal, Saraansh, et al. "Deep Learning Convolutional Neural Network for Apple Leaves Disease Detection." *International Conference Sustainable Computing in Science, Technology & Management (SUSCOM-2019)*, 2019.
- [11] Wicaksono, Guntur, et al. "Aplikasi Pendeteksi Penyakit Pada Daun Tanaman Apel Dengan Metode Convolutional Neural Network." *Information Technology and Computer Science*, vol. 5 No. 1, 2020,

- <http://publishing-widyagama.ac.id/ejournal-v2/index.php/jointecs>. Diakses pada 02 Oktober 2020.
- [12] Zhong, Yong, and Ming Zhao. "Research on deep learning in apple leaf disease recognition." *Computers and Electronics in Agriculture*, vol. 168, 2020.
- [13] Thapa, Ranjita, et al. "The Plant Pathology 2020 challenge dataset to classify foliar disease of apples." *Computer Vision and Pattern Recognition*, 2020, <https://arxiv.org/abs/2004.11958>. Diakses pada 22 Oktober 2020.
- [14] F. Chollet, "Building powerful image classification models using very little data," 2016. [Online]. Available: <https://blog.keras.io/buildingpowerful-image-classification-models-using-very-little-data.html>. Diakses pada 22 Oktober 2020.
- [15] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *Proceedings of 2017 International Conference on Engineering and Technology*, ICET 2017, 2018, doi: 10.1109/ICEngTechnol.2017.8308186.
- [16] Q. Zhang, M. Zhang, T. Chen, Z. Sun, Y. Ma, and B. Yu, "Recent advances in convolutional neural network acceleration," *Neurocomputing*, vol. 323, pp. 37–51, 2019.
- [17] J. Guérin, O. Gibaru, S. Thiery, and E. Nyiri, "CNN Features are also Great at Unsupervised Classification," pp. 83–95, 2018, doi: 10.5121/csit.2018.80308
- [18] Nadia Ramadhani, Janson Hendryli, dan Dyah Erny Herwindiati. Pencarian Objek Wisata Bersejarah di Pulau Jawa Menggunakan Convolutional Neural Network. *Jurnal Ilmu Komputer dan Sistem Operasi*. Universitas Tarumanagara. Jakarta. Diakses pada 22 Oktober 2020.
- [19] Ezz El-Din Hemdan, Marwa A. Shouma, Mohamed Esmail Karar, IEEE Member. CovidX-Net: A Framework of Deep Learning Classifiers to Diagnose Covid-19 in X-Ray Images. Available: <https://arxiv.org/abs/2003.11055>. Diakses pada 22 Oktober 2020
- [20] Szegedy, Christian et al. "Rethinking the Inception Architecture for Computer Vision." *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016): 2818-2826.
- [21] Aditya Yanuar. Inception Network. Available: <http://machinelearning.mipa.ugm.ac.id/2018/08/10/inception-network/>. Diakses pada 22 Oktober 2020
- [22] Hidayat, Ardi, et al. "DETECTION OF DISEASE ON CORN PLANTS USING CONVOLUTIONAL NEURAL NETWORK METHODS." *Jurnal Ilmu Komputer dan Informasi*, vol. 12, no. 1, 2019, pp. 51-56.
- [23] V. Bushaev, "Adam — latest trends in deep learning optimization.," <https://towardsdatascience.com/adam-latesttrends-in-deep-learning-optimization-6be9a291375c>. Diakses pada 22 Oktober 2020
- [24] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, pp. 1–15, 2015.
- [25] Categorical Cross Entropy [Online]. Available: <https://peltarion.com/knowledge-center/documentation/modeling-view/build-an-ai-model/loss-functions/categorical-crossentropy>. Diakses pada 23 Oktober 2020
- [26] S. Ghoneim, "Accuracy, Recall, Precision, F-Score & Specificity, which to optimize on?," [Online] <https://towardsdatascience.com/accuracy-recall-precision-f-scorespecificity-which-to-optimize-on-867d3f11124>. Diakses pada 23 Oktober 2020.