

TESIS

**ANALISIS KOMPARATIF AUGMENTASI DATA HAMA PENYAKIT
DAUN APEL MENGGUNAKAN ALGORITMA CONVOLUTIONAL
NEURAL NETWORK**



Disusun oleh:

Nama : Dede Husen
NIM : 21.55.1068
Konsentrasi : Digital Transformation Intelligence

PROGRAM STUDI S2 TEKNIK INFORMATIKA
PROGRAM PASCASARJANA UNIVERSITAS AMIKOM YOGYAKARTA
YOGYAKARTA

2022

TESIS

**ANALISIS KOMPARATIF AUGMENTASI DATA HAMA PENYAKIT
DAUN APEL MENGGUNAKAN ALGORITMA CONVOLUTIONAL
NEURAL NETWORK**

**COMPARATIVE ANALYSIS OF DATA AUGMENTATION ON APPLE
LEAVES DISEASE PEST USING CONVOLUTIONAL NEURAL
NETWORK ALGORITHM**

Diajukan untuk memenuhi salah satu syarat memperoleh derajat Magister



Disusun oleh:

Nama : Dede Husen
NIM : 21.55.1068
Konsentrasi : Digital Transformation Intelligence

**PROGRAM STUDI S2 TEKNIK INFORMATIKA
PROGRAM PASCASARJANA UNIVERSITAS AMIKOM YOGYAKARTA
YOGYAKARTA**

2022

HALAMAN PENGESAHAN

**ANALISIS KOMPARATIF AUGMENTASI DATA HAMA PENYAKIT DAUN
APEL MENGGUNAKAN ALGORITMA CONVOLUTIONAL NEURAL
NETWORK**

**COMPARATIVE ANALYSIS OF DATA AUGMENTATION ON APPLE
LEAVES DISEASE PEST USING CONVOLUTIONAL NEURAL NETWORK
ALGORITHM**

Dipersiapkan dan Disusun oleh

Dede Husen

21.55.1068

Telah Diujikan dan Dipertahankan dalam Sidang Ujian Tesis
Program Studi S2 Teknik Informatika
Program Pascasarjana Universitas AMIKOM Yogyakarta
pada hari Selasa, 6 Desember 2022

Tesis ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Magister Komputer

Yogyakarta, 6 Desember 2022
Rektor

Prof. Dr. M. Suyanto, M.M.
NIK. 190302001

HALAMAN PERSETUJUAN

**ANALISIS KOMPARATIF AUGMENTASI DATA PENYAKIT DAUN APEL
MENGUNAKAN ALGORITMA CONVOLUTIONAL NEURAL NETWORK**

**COMPARATIVE ANALYSIS OF DATA AUGMENTATION ON APPLE LEAVES
DISEASE PEST USING CONVOLUTIONAL NEURAL NETWORK ALGORITHM**

Dipersiapkan dan Disusun oleh

Dede Husen

21.55.1068

Telah Diujikan dan Dipertahankan dalam Sidang Ujian Tesis
Program Studi S2 Teknik Informatika
Program Pascasarjana Universitas AMIKOM Yogyakarta
pada hari Selasa, 6 Desember 2022

Pembimbing Utama

Prof. Dr. Kusriani, M.Kom
NIK. 190302106

Anggota Tim Penguji

Dhani Ariatmanto, M.Kom, Ph.D.
NIK. 190302197

Pembimbing Pendamping

Kusnawi, S.Kom., M.Eng.
NIK. 190302112

Hanafi, S.Kom., M.Eng., Ph.D
NIK. 190302024

Prof. Kusriani, M.Kom.
NIK. 190302106

Tesis ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Magister Komputer

Yogyakarta, 6 Desember 2022
Direktur Program Pascasarjana

Prof. Dr. Kusriani, M.Kom.
NIK. 190302106

HALAMAN PERNYATAAN KEASLIAN TESIS

Yang bertandatangan di bawah ini,

Nama mahasiswa : Dede Husen
NIM : 21.55.1068
Konsentrasi : Digital Transformation Intelligence

Menyatakan bahwa Tesis dengan judul berikut:
**Analisis Komparatif Augmentasi Data Penyakit Daun Apel Menggunakan
Algoritma Convolutional Neural Network**

Dosen Pembimbing Utama : Prof. Dr. Kusriani, M.Kom
Dosen Pembimbing Pendamping : Kusnawi, S.Kom., M.Eng.

1. Karya tulis ini adalah benar-benar ASLI dan BELUM PERNAH diajukan untuk mendapatkan gelar akademik, baik di Universitas AMIKOM Yogyakarta maupun di Perguruan Tinggi lainnya
2. Karya tulis ini merupakan gagasan, rumusan dan penelitian SAYA sendiri, tanpa bantuan pihak lain kecuali arahan dari Tim Dosen Pembimbing
3. Dalam karya tulis ini tidak terdapat karya atau pendapat orang lain, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan disebutkan dalam Daftar Pustaka pada karya tulis ini
4. Perangkat lunak yang digunakan dalam penelitian ini sepenuhnya menjadi tanggung jawab SAYA, bukan tanggung jawab Universitas AMIKOM Yogyakarta
5. Pernyataan ini SAYA buat dengan sesungguhnya, apabila di kemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka SAYA bersedia menerima SANKSI AKADEMIK dengan pencabutan gelar yang sudah diperoleh, serta sanksi lainnya sesuai dengan norma yang berlaku di Perguruan Tinggi

Yogyakarta, 6 Desember 2022

Yang Menyatakan,



The image shows a handwritten signature in black ink over a rectangular stamp. The stamp contains the logo of Universitas AMIKOM Yogyakarta, the text 'UNIVERSITAS AMIKOM YOGYAKARTA', and the identification number '156798510'. The signature is written in a cursive style.

Dede Husen

HALAMAN PERSEMBAHAN

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Alhamdulillah atas izin Allah SWT penelitian tesis ini selesai, suatu kebahagiaan serta kebanggaan bisa ada pada titik ini, tesis ini saya persembahkan untuk Ibunda, ayahanda (almarhum), kakak, adik serta keluarga kecil kami terutama sang istri yang selalu berada di samping mendampingi di setiap revisi dengan teh tarik yang selalu tersedia setiap pagi dan malam disamping laptop.

Tak lupa tesis ini juga saya persembahkan juga kepada almamater S1 tercinta FKOM Universitas Kuningan khususnya Prodi Teknik Informatika yang telah memberikan kepercayaan untuk menerima beasiswa S2 alumni berprestasi perdana.

Terakhir penelitian ini juga saya persembahkan untuk almamater tercinta, tempat bertumbuh, belajar dan berkembang Universitas Amikom Yogyakarta serta para pembaca yang budiman semoga penelitian ini dapat bermanfaat serta dapat dikembangkan lebih baik lagi.

HALAMAN MOTTO

"Tiada Prestasi Tanpa Disiplin".



"Barang siapa belum pernah merasakan pahitnya mencari ilmu walau sesaat, ia akan menelan hinanya kebodohan sepanjang hidupnya."

- Imam Syafi'i-

"Hal yang paling di sesalkan seorang manusia ketika masa tua menghampiri ialah tidak pernah mencoba sama sekali merasakan kegagalan"

-Anonim-

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Alhamdulillah, senantiasa kita panjatkan puji dan syukur kepada sang khaliq, Allah *subhanallah wata'ala* yang telah memberikan segala karunia dan nikmat yang tak pernah habis hingga saat ini, sehingga tesis yang berjudul “**Analisis Komparatif Augmentasi Data Penyakit Daun Apel Menggunakan Algoritma Convolutional Neural Network**” telah selesai dengan tepat waktu tentu berkat nikmat dan ridho yang telah diberikan oleh-Nya.

Melalui halaman ini, izinkan Penulis untuk menyampaikan ungkapan terimakasih yang tak terhingga kepada:

1. Kedua Orang Tua Ayahanda (almarhum) dan Ibunda yang senantiasa memberikan Do'a dan dukungan terbaiknya.
2. Istri tercinta Dede Ulfah Nurlulu yang senantiasa menemani setiap detik sepanjang masa studi ini
3. Bapak Prof. Dr. Suyanto, MM, selaku Rektor Universitas AMIKOM Yogyakarta.
4. Ibu Prof. Dr. Kusriani, M.Kom, selaku Direktur Pascasarjana Universitas Amikom Yogyakarta sekaligus Pembimbing utama yang telah memberikan bimbingan terbaiknya dan tentunya ilmu dan arahan beliau dalam penelitian ini.
5. Bapak Kusnawi, M.Eng., selaku Dosen pembimbing pendamping yang telah memberikan masukan-masukan terkait penelitian ini.
6. Bapak Dhani Ariatmanto, M.Kom, Ph.D. selaku penguji utama pada ujian tesis yang telah memberikan arahan, masukan serta perbaikan agar penelitian ini menjadi lebih baik
7. Bapak Hanafi, S.Kom., M.Eng., Ph.D. selaku penguji kedua pada ujian tesis yang telah banyak memberikan pandangan serta masukan-masukan yang membangun.

8. Segenap Dosen dan Staff Magister Teknik Informatika Universitas AMIKOM Yogyakarta yang telah memberikan ilmu, wawasan, bantuan, pengalaman hingga petuah-petuah selama masa kuliah.
9. Rekan-rekan seperjuangan Magister Teknik Informatika PJJ dari sabang sampai merauke khususnya kawan-kawan konsentrasi *Digital Transformation Intelligence* yang senantiasa saling bahu membahu dalam menyelesaikan tugas-tugas yang hadir menemani setiap pekannya.
10. Terakhir tentu kepada diri sendiri yang telah berjuang sampai pada titik ini, melawan kemalasan dan ketidakmungkinan yang menjadi nyata.

Yogyakarta, 6 Desember 2022

Penulis



DAFTAR ISI

HALAMAN PENGESAHAN.....	iii
HALAMAN PERSETUJUAN.....	iv
HALAMAN PERNYATAAN KEASLIAN TESIS.....	v
HALAMAN PERSEMBAHAN.....	vi
HALAMAN MOTTO.....	vii
KATA PENGANTAR.....	viii
DAFTAR ISI.....	x
DAFTAR TABEL.....	xiii
DAFTAR GAMBAR.....	xiv
DAFTAR LISTING.....	xvi
INTISARI.....	xvii
<i>ABSTRACT</i>	xviii
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang Masalah.....	1
1.2. Rumusan Masalah.....	5
1.3. Batasan Masalah.....	6
1.4. Tujuan Penelitian.....	7
1.5. Manfaat Penelitian.....	7
BAB II TINJAUAN PUSTAKA.....	8
2.1. Tinjauan Pustaka.....	8
2.2. Keaslian Penelitian.....	12

2.3. Landasan Teori.....	20
2.3.1. Hama Tanaman.....	20
2.3.2. <i>Convolutional Neural Network</i>	23
2.3.2.1. <i>Convolutional Layers</i>	24
2.3.2.2. <i>Pooling Layers</i>	24
2.3.2.3. <i>Rectified Linear Units Layer (ReLu Layer)</i>	25
2.3.2.4. <i>Flattening</i>	25
2.3.2.5. Dropout Regularization.....	26
2.3.2.6. Fully Connected Layer.....	27
2.3.2.7. SoftMax.....	27
2.3.3. Augmentasi Data.....	28
2.2.3.1. Augmentasi Data Berdasarkan Kelengkungan Data (<i>Data Waring</i>).....	29
2.2.3.2. Systemic Over.....	31
BAB III METODE PENELITIAN	33
3.1. Jenis, Sifat, dan Pendekatan Penelitian.....	33
3.1.1. Jenis Penelitian.....	33
3.1.2. Sifat Penelitian.....	33
3.1.3. Pendekatan Penelitian.....	33
3.2. Metode Pengumpulan Data.....	34
3.3. Metode Analisis Data.....	34

3.4. Alur Penelitian	35
BAB IV HASIL PENELITIAN DAN PEMBAHASAN	37
4.1. Analisis Kebutuhan Data	37
4.2. Data Citra Daun Apel.....	37
4.3.1 Pengujian Metode CNN dengan Dataset Asli.....	41
4.3.2 Pengujian Metode CNN dengan Teknik Augmentasi Data	45
4.3. Evaluasi Model.....	74
4.3.1. Pengaruh Ukuran Citra	75
4.3.2. Pengaruh Jumlah <i>Epoch</i>	75
4.3.3. Perbandingan dengan penelitian sebelumnya	77
BAB V PENUTUP	78
5.1. Kesimpulan	78
5.2. Saran	79
DAFTAR PUSTAKA	80
LAMPIRAN	86

DAFTAR TABEL

Table 2.1. Matriks Literatur Review Dan Posisi Penelitian Sistem Cerdas Deteksi Hama Pada Daun Apel Menggunakan Algoritma <i>Convolutional Neural Network</i>	12
Table 4.1. Sebaran Dataset Asli	42
Table 4.2. Tabel Prediksi Model CNN Dataset Asli	43
Tabel 4.3. Prediksi Model CNN Dengan Augmentasi Data <i>Horizontal Flip</i>	51
Tabel 4.4. Prediksi Model CNN Dengan Augmentasi Data <i>Vertical Flip</i>	59
Tabel 4.5. Tabel Rumus Dan Persamaan Transformasi Rotasi.....	61
Tabel 4.6. Tabel Prediksi Model CNN Dengan Augmentasi <i>Random Rotation</i>	66
Tabel 4.7. Tabel Prediksi Model CNN Dengan Augmentasi Data Gabungan	73
Tabel 4.8. Tabel Perbandingan <i>Accuracy & Loss</i> Berdasarkan Ukuran Citra.....	75
Tabel 4.9. Perbandingan Akurasi Dan <i>Loss</i> Berdasarkan Jumlah <i>Epoch</i>	76

DAFTAR GAMBAR

Gambar 2.1. Gambar Hama <i>Scab</i> Pada Daun Apel	21
Gambar 2.2. Hama <i>Black Rot</i> Pada Daun Apel.....	22
Gambar 2.3. Hama <i>Cedar/Rust</i> Pada Daun Apel.....	23
Gambar 2.4. Arsitektur <i>Convolutional Neural Network</i>	24
Gambar 2.5. Contoh Implementasi <i>Average Pooling & Max Pooling</i>	25
Gambar 2.6. Grafik fungsi aktivasi pada <i>ReLu Layer</i>	25
Gambar 2.7. Contoh Proses <i>flattening</i>	26
Gambar 2.8. <i>Standard Network</i> (kiri) dan <i>Network dengan dropout regulation</i>	27
Gambar 2.9. Pengelompokan Jenis Augmentasi Data.....	28
Gambar 2.10. Contoh Augmentasi Data.....	29
Gambar 2.11. Contoh Implementasi <i>Adversarial Training</i>	30
Gambar 2.12. Contoh Penerapan <i>Neural Style Transfer</i>	31
Gambar 3.1. Flowchart Penelitian.....	36
Gambar 4.1. Daun Apel Sehat (a), hama daun <i>Apple Scab</i> (b), hama daun <i>Apple Black Rot</i> (c), hama daun <i>Apple Cedar</i> (d).....	38
Gambar 4.2. <i>Accuracy & Loss</i> CNN Dengan Dataset Asli.....	42
Gambar 4.3. <i>Confusion Matrix</i> Model CNN Dengan Dataset Asli.....	44
Gambar 4.4. Contoh Transformasi Citra Secara Horizontal	45
Gambar 4.5. Matriks Transformasi horizontal (Pencerminan).....	48
Gambar 4.6. Ilustrasi Matrik Transformasi refleksi <i>horizontal</i>	47
Gambar 4.7. <i>Accuracy & Loss</i> Model CNN Augmentasi Horizontal Flip.....	50

Gambar 4.8. <i>Confusion Matrix</i> Model CNN Augmentasi Horizontal Flip.....	52
Gambar 4.9. Contoh transformasi citra secara vertikal (<i>vertical flip</i>).....	53
Gambar 4.10. Matriks transformasi vertikal (<i>vertical flip</i>).....	54
Gambar 4.11. <i>Accuracy & Loss</i> Model CNN Augmentasi Vertical Flip.....	55
Gambar 4.12. <i>accuracy & loss</i> model CNN teknik augmentasi vertical flip	58
Gambar 4.13. <i>Confusion Matrix</i> Model CNN Augmentasi <i>Vertical Flip</i>	60
Gambar 4.14. Contoh Transformasi Rotasi	61
Gambar 4.15. <i>Accuracy & Loss</i> Model CNN Augmentasi <i>Random Rotation</i>	66
Gambar 4.16. <i>Confusion Matrix</i> Model CNN Augmentasi <i>Random Rotation</i>	68
Gambar 4.17. <i>Accuracy & Loss</i> Model CNN Teknik Augmentasi Gabungan... ..	73
Gambar 4.18. <i>Confusion Matrix</i> Model CNN Augmentasi <i>Random Rotation</i>	74

DAFTAR LISTING

Listing 4.1 <i>Source Code</i> tahapan inialisasi Input Model CNN	41
Listing 4.2 <i>Source Code Training</i> Model CNN dengan Augmentasi Data <i>Horizontal Flip</i>	49
Listing 4.3 <i>Source Code Training</i> Model CNN dengan Augmentasi Data <i>Vertical Flip</i>	57
Listing 4.4 <i>Source Code Training</i> Model CNN Augmentasi Data <i>Random Rotation</i> ...	64
Listing 4.5 <i>Source Code Training</i> Model CNN Augmentasi Data <i>Gabungan</i>	71



INTISARI

Dewasa ini kebutuhan akan konsumsi buah-buahan kian meningkat seiring dengan meningkatnya populasi manusia dan kesadaran akan konsumsi makanan bergizi, buah apel merupakan salah satu buah yang paling banyak dikonsumsi oleh manusia di seluruh dunia. Menurut data yang dikutip dari Badan Pusat Statistik Nasional tahun 2021 produksi buah apel tahun 2021 menurun dari tahun sebelumnya dari 519.531 ton menjadi 509.544 ton. Salah satu penyebab menurunnya produksi apel adalah adanya gangguan hama pada tanaman apel tersebut. Setidaknya ada beberapa jenis hama yang dapat diidentifikasi pada daun apel yaitu *Apple Scab* (*venturia inaequalis*), *Apple Black Root* (*Botryosphaeria*) dan *Apple Cedar/Rust* (*Gymnosporangium juniperi virginianae*). Tahapan penelitian dimulai dengan melakukan beberapa studi literatur seputar penelitian terkait, kemudian merumuskan dan memvalidasi masalah serta mulai melakukan pengumpulan data dari *public dataset Kaggle*. Kemudian dalam tahap eksprimennya penulis membagi dataset kedalam tiga bagian dengan persentase 80% data training, 10 % data validasi dan 10% data testing. Metode klasifikasi citra yang digunakan algoritma *Convolutional Neural Network (CNN)* untuk membuat sebuah model yang dapat mengklasifikasikan data citra, para prosesnya implementasi penulis menggunakan bahasa pemrograman python untuk membangun model tersebut. Penulis melakukan beberapa eksperimen dengan melakukan perubahan pada beberapa parameter model yang berpengaruh terhadap akurasi model. Untuk evaluasi performa dan akurasi model menggunakan *confusion matrix*. Hasil dari penelitian menunjukkan bahwa ukuran citra, augmentasi data dan jumlah *epoch* sangat berpengaruh terhadap akurasi model, dari hasil uji tersebut model *CNN* yang paling baik akurasinya adalah dengan model dengan parameter ukuran citra 256x256, dengan teknik gabungan yaitu augmentasi data *horizontal flip*, *vertical flip* dan *random rotation* serta jumlah *epoch* ke-60 memiliki tingkat akurasi tertinggi yaitu 99.57%. hasil dari penelitian ini diharapkan dapat di implementasikan dalam sebuah aplikasi yang dapat di gunakan langsung oleh para petani dalam mendeteksi hama pada tanaman apel secara cepat dan akurat.

Kata Kunci: Hama tanaman apel; Identifikasi; *Convolutional Neural Network*; Augmentasi Data; *Overfitting*

ABSTRACT

Today the need for fruit consumption is increasing along with the increasing human population and awareness of the consumption of nutritious foods, apples are one of the most consumed fruits by humans worldwide. According to data from the Indonesian National Statistics Center 2021, apple production in 2021 decreased from the previous year from 519,531 tons to 509,544 tons. One of the causes of the decline in apple production is the presence of pests in the apple plant. At least several types of pests can be identified on apple leaves, namely Apple Scrub (*Venturia inaequalis*), Apple Black Root (*Botryosphaeria*), and Apple Cedar/Rust (*Gymnosporangium juniperi virginianae*). The research stage begins with conducting several literature studies regarding related research, then formulating and validating the problem and starting to collect data from the Kaggle public dataset. Then in the experimental stage, the author divides the dataset into three parts with a percentage of 80% training data, 10% validation data and 10% testing data. The image classification method used is the Convolutional Neural Network (CNN) algorithm to create a model that can classify image data, the process of implementation author uses the python programming language to build the model. The author conducted several experiments by making changes to several model parameters that affect the accuracy of the model. To evaluate the performance and accuracy of the model using a confusion matrix. The results of the study indicate that image size, data augmentation and, the number of epochs greatly affect the accuracy of the model, from the test results the CNN model with the best accuracy is the model with the image size parameter 256x256, augmentation data combination like horizontal flip, vertical flip and, random rotation data augmentation and the number of the 60th epoch has the highest accuracy rate of 99.57%. The results of this study are expected to be implemented in an application that can be used directly by farmers in detecting pests on apple plants quickly and accurately.

Keywords: Apple Plant Disease; Identification; Convolutional Neural Network; Data Augmentation; Overfitting

BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Saat ini pertanian mempunyai peranan penting dalam kehidupan modern dan berkontribusi penuh dalam perkembangan ekonomi dan ketahanan pangan. Salah satu produk pertanian adalah buah apel. Apel merupakan salah satu buah yang paling banyak dikonsumsi di seluruh dunia yang berasal dari daerah Asia barat, buah ini dapat tumbuh di iklim subtropis dan pada tahun 1934 tanaman apel ini mulai di tanam di Indonesia. Salah satu masalah yang saat ini dihadapi oleh para petani adalah adanya hama tanaman apel. Menurut data statistik yang dikeluarkan oleh Badan Pusat Statistik Nasional tahun 2021 tentang statistik produksi buah-buahan di Indonesia buah apel di produksi adalah sebanyak 509.544 ton (Badan Pusat Statistik Nasional, 2021) lebih sedikit dari tahun sebelumnya yakni 519.531 (Badan Pusat Statistik Nasional, 2020) Beberapa jenis hama pada tanaman apel terdapat setidaknya 3 jenis diantaranya adalah jenis yaitu hama *Apple Scab* yang disebabkan oleh jamur *venturia inaequalis*, *Apple Black Rot* oleh jamur *Botryosphaeria obtusa* dan *Apple Cedar / Apple Rust* yang disebabkan oleh jamur *Gymnosporangium juniperi-virginianae*, semua hama tersebut mengakibatkan pertumbuhan tanaman dan kualitas buah yang dihasilkan kurang baik yang imbasnya adalah kepada nilai jual buah apel tersebut. Di Indonesia sendiri untuk mengendalikan masalah tersebut kebanyakan masih menggunakan cara-cara tradisional, salah satunya adalah dengan memberikan cairan pembasmi hama

seperti pestisida pada tanaman tersebut, dimana penggunaan pestisida mempunyai dampak negatif baik bagi kesehatan maupun lingkungan sekitar. Kebutuhan petani akan ahli atau pakar hama dan penyakit tanaman juga cukup diperlukan dikarnakan pakar tersebut dapat memberikan analisis terkait dengan hama apa yang ada pada pohon apel tersebut agar dapat memberikan solusi terbaik secara akurat, akan tetapi orang yang ahli di bidang tersebut tentunya masih terasa kurang. Maka hal yang paling mungkin dilakukan adalah dengan memanfaatkan teknologi yang ada dan mungkin digunakan saat ini guna membantu para petani untuk mengendalikan hama pada tanaman apel tersebut.

Dataset public hama pada daun apel yang ada adalah dataset citra digital berwarna (RGB) yang telah terklasifikasikan sesuai dengan penyakit yang ada pada tanaman tersebut, secara kasat mata kita dapat melihat perbedaan antara ke empat jenis hama pada daun tersebut berdasarkan perbedaan warna pada daun tersebut.

Jika dilihat karakteristik daun yang menyukai hama memiliki karakteristik tertentu, seperti dari warna bercak pada daun masing-masing hama memiliki ciri-ciri tertentu. Dalam pengolahan citra digital khususnya algoritma *convolutional neural network* akan melihat sebuah gambar sebagai kumpulan angka-angka dalam sebuah matriks, dimana citra digital berwarna dapat di representasikan dengan tiga buah matriks, satu matriks merah, satu matriks hijau dan matriks biru atau yang dikenal dengan matriks RGB. Kemudian perbedaan sebaran warna pada setiap citra akan menjadi segmentasi tertentu dalam proses training data.

Berdasarkan permasalahan di atas kita dapat melakukan pendekatan digital yakni dengan membuat sebuah aplikasi berbasis *mobile* yang secara cerdas dapat

mendeteksi jenis hama secara akurat, cepat dan otomatis berdasarkan data-data yang telah terkumpul. Teknologi kecerdasan buatan (*artificial intelligence*) dapat menjadi solusi bagi masyarakat dalam mengembangkan sistem pertanian modern agar lebih efektif, akurat dan efisien khususnya kecerdasan buatan dengan model *deep learning*. selain itu *deep learning* dirancang untuk terus menerus menganalisis data yang masuk dengan bantuan salah satu algoritma *Artificial Neural Network (ANN)* yaitu *Convolutional Neural Network (CNN)* yang biasa digunakan untuk menganalisis data berbentuk citra atau gambar.

Dengan metode *Convolutional Neural Network* ini pata peneliti secara empiris telah terbukti mampu mengklasifikasikan citra dengan baik seperti yang dilakukan Sumita Misrah dkk. (Mishra dkk., 2020) dalam penelitiannya yang menerapkan model *Deep CNN* untuk mengklasifikasikan jenis hama tanaman pada daun jagung dengan tingkat akurasi deteksi tertinggi sebanyak 88.4%. Penelitian serupa juga telah dilakukan oleh Abdu Jalil dkk. (Rozaqi dkk., 2021). menggunakan metode *Convolutional Neural Network* pada daun kentang yang telah terklasifikasi berdasarkan jenis penyakitnya.

Untuk membuat sebuah model algoritma yang mempunyai nilai akurasi yang tinggi dengan metode *Convolutional Neural Network* maka di perlukan jumlah data yang banyak terlebih citra yang di ambil juga harus terklasifikasi terlebih dahulu. Beberapa penelitian menyebutkan bahwa untuk membuat sebuah model itu mempunyai akurasi tinggi dan mengurangi *overfitting* pada model, salah satunya adalah penelitian yang dilakukan oleh (Enkvetchakul & Surinta, 2021), menyebutkan dalam penelitiannya dengan melakukan experiment pada ribuan

dataset penyakit daun pada tanaman dimana pada augmentasi data beberapa teknik kombinasi dilakukan yang menghasilkan 15 dataset baru dari setiap 1 data asli dan berhasil meningkatkan akurasi menjadi 76,15% dari dataset yang di training dengan tanpa melakukan augmentasi data yang nilai akurasinya hanya 63,08%. Pada penelitian lain yang juga dilakukan oleh Olusola Oluwakemi Abayomi dkk (Abayomi-Alli dkk., 2021) pada penelitiannya ojek deteksi pada daun singkong dengan menggunakan arsitektur MobileNetV2, dimana pada penelitian tersebut berfokus meningkatkan kualitas dataset dengan kualitas rendah dimana di terapkan 4 metode yaitu metode *resolution down-sampling*, *Gaussian Blurring*, *Motion Blur* dan *over exposure* untuk mereduksi kualitas gambar, sehingga dataset training yang digabungkan mampu meningkatkan akurasi hingga 3% - 15%.

Teknik augmentasi data sangatlah beragam, dalam hal ini yang menjadi fokus pada riset ini adalah bagaimana menentukan metode augmentasi yang tepat pada dataset hama daun apel ini, dikutip dari towarddatascience.com setidaknya terdapat 3 hal yang harus di perhatikan dalam menentukan teknik augmentasi data yang tepat, yang pertama adalah *domain expertise* dimana penerapan data augmentasi harus di sesuaikan dengan tujuan akhir projeknya, dimana pada beberapa kasus penerapan augmentasi data justru malah membuat data tambahan yang tidak relevan, yang kedua adalah kebutuhan bisnis, seperti contoh jika tujuan kita memperbanyak data untuk kebutuhan *computer vision self driving car* apakah perlu menerapkan teknik augmentasi data secara *vertical flip*, karna jika tidak diterapkan maka jika model system yang ada pada data tersebut tidak akan mengenal citra terbalik dan mungkin akan terjadi error atau bahkan kegagalan sistem, yang ketiga

adalah jika terlalu banyak teknik augmentasi yang diterapkan maka akan berpengaruh kepada beban komputasi dan akan banyak citra yang tidak dikenali secara normal sehingga banyak data yang tidak dipakai karena maksud dari citra tersebut sudah tidak ada (Olga Chernytska, t.t.). Pada penelitian ini penulis akan melakukan beberapa eksperimen untuk mencari kombinasi antara algoritma CNN dan teknik augmentasi yang tepat, dimana hasil dari model tersebut diharapkan dapat mendeteksi hama pada daun apel.

berdasarkan studi literatur yang telah dilakukan pada beberapa kasus di atas penulis akan memakai *dataset public* penyakit daun apel dengan melakukan teknik-teknik augmentasi data pada dataset daun apel yang ada, dan metode yang akan digunakan pada penelitian ini adalah dengan menggunakan algoritma *Convolutional Neural Network* serta akan melakukan beberapa teknik augmentasi data dan membandingkan teknik augmentasi mana yang mempunyai tingkat akurasi yang paling baik dan tentunya dengan *overfitting* yang rendah.

1.2. Rumusan Masalah

Berdasarkan uraian pada latar belakang di atas, maka penulis dapat merumuskan masalahnya ke dalam beberapa poin berikut:

- a. Bagaimana cara membangun sebuah model CNN yang di padukan dengan augmentasi data yang tepat sehingga menghasilkan akurasi yang tinggi untuk mendeteksi hama pada daun apel?

- b. Berapakah tingkat akurasi yang di hasilkan ketika menggunakan metode *Convolutional Neural Network* sebelum dan sesudah dilakukan augmentasi data?
- c. Teknik augmentasi data yang manakah yang memiliki tingkat akurasi tertinggi berdasarkan studi kasus pada citra daun apel?

1.3. Batasan Masalah

Agar penelitian fokus pada pembahasan maka penulis memberikan batasan-batasan masalah sebagai berikut :

- a. Penelitian menggunakan dataset publik yang diperoleh dari Kaggle
- b. Objek deteksi yang digunakan adalah citra daun apel yang telah mempunyai 4 label yaitu *healthy, multiple diseases, cedar/rust* dan *scab*.
- c. Algoritma yang digunakan pada penelitian ini adalah dengan menggunakan algoritma *Convolutional Neural Network*
- d. *Output* yang dihasilkan adalah berupa hasil klasifikasi jenis hama pada daun apel tersebut
- e. Pengolahan data, pembuatan model akan dilakukan dengan menggunakan Bahasa pemrograman python dan *library* *tensor flow* yang semuanya akan dijalankan melalui *tools* pembantu *jupyter notebook*.
- f. Pada penelitian ini akan dilakukan beberapa teknik augmentasi data tranformasi geometris.

1.4. Tujuan Penelitian

Berikut tujuan penelitian yang diharapkan oleh penulis:

- a. Mengembangkan Algoritma CNN yang dipadukan dengan teknik-teknik augmentasi yang tepat untuk meningkatkan performa akurasi pada model yang dibangun untuk mendeteksi penyakit pada daun apel.
- b. Menentukan teknik data augmentasi yang tepat untuk menghindari *overfitting* pada model yang di bangun
- c. Membantu para petani dalam mengendalikan hama pada tanaman apel secara cepat dan akurat.
- d. Meningkatkan akurasi model yang di bangun agar model mampu melakukan deteksi citra daun apel dan mengklasifikasinya dengan akurat

1.5. Manfaat Penelitian

Berikut manfaat dari penelitian ini.

- a. Implementasi teknologi ini diharapkan dapat menjadi investasi penting dalam pengendalian hama dibidang pertanian khususnya di Indonesia.
- b. Bagi para petani, implementasi teknologi ini dapat membantu dalam meningkatkan kualitas buah dan meminimalisir hama.
- c. Dengan mengimplementasikan sistem cerdas ini tentunya akan membuka peluang riset lebih jauh lagi baik para peneliti baik dibidang agrikultur maupun dibidang teknologi, dan tentunya akan melahirkan sebuah kombinasi diantara keduanya yang hasilnya akan banyak membantu para petani.

BAB II

TINJAUAN PUSTAKA

2.1. Tinjauan Pustaka

Beberapa penelitian telah dilakukan terkait masalah tema ini, seperti yang dilakukan oleh (Mishra dkk., 2020) dalam pada penelitian yang berjudul “Kecerdasan buatan dalam pertanian untuk deteksi cacat pada buah apel menggunakan algoritma CNN” ini penulis melakukan pemodelan menggunakan algoritma *Convolutional Neural Network* dengan metode *backpropagation* untuk mengklasifikasikan buah apel ke dalam 4 jenis label yaitu apel yang mempunyai kondisi baik (*good apple*), apel dengan kondisi buruk atau tidak layak dikonsumsi (*Damage Apple*), setelah melakukan *training* data penulis menguji sistem yang telah di *training* dan menghasilkan akurasi 87,5% untuk skema label *good apple* dan 50% untuk skema *damage apple*.

Dalam penelitiannya yang berjudul “Klasifikasi Penyakit Tanaman apel dari citra daun dengan *convolutional neural network*”(Asrafil dkk., 2021), penulis melakukan pemodelan terhadap 7700 *dataset training* dan 1943 data validasi yang dibagi kedalam 4 kelas yaitu *Apple scab*, *black root*, *rust* dan *healthy*, sebelum proses *training* semua data yang masuk akan di lakukan proses *wrapping* dan *cropping* sesuai dengan *edge* objek utama dengan dimensi 256x256 piksel dengan format RGB sehingga diharapkan akan menghasilkan akurasi tinggi dan tentunya menghilangkan *noise*. Pada tahapan uji coba masing-masing dari tiap kelas telah

di uji dengan tingkat akurasi tinggi, dan hasil akurasi keseluruhan model *training* tersebut adalah sebesar 97.1% akurasi.

Dalam penelitian lain yang berjudul “Aplikasi Pendeteksi Penyakit Pada Daun Tanaman Apel Dengan Metode *Convolutional Neural Network*” (Wicaksono & Andryana, 2018), arsitektur LeNet-5 digunakan dalam metode *Convolutional Neural Network* dengan mengolah 3151 dataset citra daun apel dengan tingkat akurasi 75%, peneliti disini menggunakan *dataset* yang berasal dari *Plant Village Crowd AI*. Penelitian ini menggunakan model arsitektur LeNet-5 dengan mengubah parameter agar meningkatkan hasil akurasi, yaitu menggunakan 32 *maps* yang berukuran 5 x 5 (C1) dengan *maxpooling* 2 x 2 (S1). Selanjutnya menggunakan 64 *maps* yang berukuran 5 x 5 (C2) dengan *maxpooling* 2 x 2 (S2). Kemudian dilanjutkan dengan memakai 2 *hidden layer*, *hidden layer* pertama memiliki 280 neuron dan pada *hidden layer* kedua memiliki 150 *neuron*. Kemudian untuk mengetahui tingkat akurasi model dengan menggunakan *confusion matrix recall*. Pengujian ini dilakukan dengan melakukan iterasi *epoch* yaitu *epoch* 50, *epoch* 75 dan 100 dimana *epoch* tertinggi memiliki tingkat akurasi tertinggi yaitu 97.8%. Hasil dari *confusion matrix* didapatkan hasil akhir akurasi model sebesar 89.62%.

Metode *Convolutional Neural Network* juga di pakai pada penelitian yang berjudul “Penggunaan *Deep Learning* Dengan Metode CNN untuk Klasifikasi Kualitas Sayur Kol berdasarkan Citra Fisik”(Dhamayanti dkk., 2021) Pada penelitian ini penulis menguji 270 *dataset* sayur kol yang di kelompokkan ke dalam 2 label yaitu sayur kol yang baik/layak dan buruk serta untuk pembagiannya 210 *data training* dan 60 *data testing*. Pada penelitian ini penulis menggunakan skema

dengan *learning rate* sebesar 0.004, iterasi *epoch* 30 dan 3 metode *Stochastic Gradient Descent* (SGD), *Adaptive moment* (Adam), dan *Root Mean Square Propagation* (RMSprop) dengan hasil akurasi tertinggi pada metode *Adaptive Moment* (Adam) dengan akurasi *data testing* 80% dan 73% *data training* berdasarkan komposisi warna yang terdapat dalam citra.

Dalam artikel publikasinya yang berjudul "*Identification of Apple Leaf Diseases Based on Deep Convolutional Neural Networks*" pada penelitiannya penulis mengumpulkan 13.689 *dataset* daun apel yang telah di kelompokkan ke dalam 4 jenis penyakit daun apel yaitu *Mosaic*, *Rust*, *Brown Spot* dan *Alternaria* digunakan untuk membuat model untuk mendeteksi penyakit pada daun apel, metode yang digunakan pada penelitian ini adalah dengan menggunakan metode *Deep Convolutional Neural Network* (Liu dkk., 2018).

Hasil eksperimen menunjukkan bahwa model yang di buat pada penelitian ini mencapai tingkat akurasi tinggi yakni sebesar 97,62 % pada data *testing* dan tentunya lebih tinggi dari model tradisional lainnya, jika dibandingkan dengan model AlexNet standar, parameter yang di gunakan berkurang secara signifikan yaitu sebesar 51.206.928 dan menunjukkan tingkat konvergensi yang lebih cepat. Kemudian metode yang dihasilkan terbukti lebih tinggi akurasinya yakni sebesar 10,83% jika menggunakan *dataset* sintesis dibanding dengan *dataset* asli.

Pada penelitian ini penulis melakukan duplikasi data gambar yang asli, dimana data gambar asli hanya terdapat 1053 gambar masing-masing 252 gambar *mosaic*, 319 gambar *Brown Spot*, 182 Gambar *Rust*, dan 300 gambar *Alternaria leaf spot* dimana semua gambar tersebut diambil di beberapa daerah di China dengan

resolusi asli 2456x2058 piksel. Teknik yang digunakan dalam memperbanyak *dataset* ini adalah dengan melakukan metode *image processing direction disturbance* yakni dengan mengubah arah/angel dari gambar tersebut dengan rotasi 90°, 180°, 270° dan *mirror symmetry* termasuk *horizontal symmetry* kemudian yang kedua adalah metode *Light Disturbance* dengan melakukan beberapa pemrosesan pada gambar yaitu proses *low brightness*, *high brightness*, *low contrast*, *high contrast*, *low sharpness* dan *high sharpness*. Pada akhirnya dari sebuah gambar asli akan menjadi 12 *dataset*, hasil akhir dari *image processing* ini adalah terdapat 1088 *data training* dan 2801 gambar *testing* telah terbentuk dengan dimensi baru sebesar 256x256 piksel.

Penelitian serupa juga pernah dilakukan oleh (Rozaqi dkk., 2021) dengan daun kentang sebagai objek penelitiannya. *Dataset training* yang terdiri dari 922 citra daun kentang yang telah di klasifikasikan dengan rincian 400 daun kentang dengan kelas *Later Bright*, 400 daun kentang dengan kelas *Early Bright* dan 122 Daun dengan kelas *Healthy*. Pada *epoch* ke-10 dengan *batch size* 20 dengan total *data training* 922 gambar dan *data testing* 230 gambar menghasilkan nilai akurasi 95% dan untuk akurasi validasi menghasilkan 94%.

2.2. Keaslian Penelitian

Table 2.1. Matriks literatur review dan posisi penelitian
Sistem Cerdas Deteksi Hama Pada Daun Apel Menggunakan Algoritma *Convolutional Neural Network*

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
1	Kecerdasan buatan dalam pertanian untuk deteksi cacat pada buah apel menggunakan algoritma <i>Convolutional Neural Network (CNN)</i>	Linggar Kiki Zakaria, Irwan Krisnandi, 2020	Mengklasifikasikan kualitas buah apel berdasarkan citra dengan menggunakan algoritma CNN	Pada penelitian ini penulis melakukan pemodelan menggunakan algoritma <i>Convolutional Neural Network</i> dengan metode <i>backpropagation</i> untuk mengklasifikasikan buah apel ke dalam 4 jenis label yaitu apel yang mempunyai kondisi baik (<i>good apple</i>), apel dengan kondisi buruk atau tidak layak dikonsumsi (<i>Damage Apple</i>), setelah melakukan <i>training</i> data penulis menguji sistem yang telah di <i>training</i> dan menghasilkan akurasi 87,5% untuk skema label <i>good apple</i> dan 50 % untuk skema <i>Damage apple</i> .	Pada penelitian ini dataset yang digunakan sedikit, tidak dilakukan teknik augmentasi data untuk memperbanyak dataset, sehingga akurasi masih diragukan.	Pada penelitian yang saya lakukan, dataset terdiri lebih dari 3500 data citra daun apel yang telah di labeli sesuai klasifikasinya, serta menerapkan teknik augmentasi data untuk memperbanyak dataset.

Tabel 2.1 (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
1	Klasifikasi Penyakit Tanaman apel dari citra daun dengan <i>convolutional neural network</i>	Andir Asrafil Ardan Paliwang, M Ridwan Dwi Septian, Margi Cahyati, Erick Rachmat Swedia. - 2020 - SEBATIK 2020	Mengklasifikasikan penyakit tanaman apel melalui deteksi Citra daun apel menggunakan cnn Daun Apel	Dalam penelitiannya, penulis melakukan pemodelan terhadap 7700 dataset training dan 1943 data Validasi yang dibagi kedalam 4 kelas yaitu <i>Apple scab, black root, rust</i> dan <i>healthy</i> , sebelum proses training semua data yang masuk akan di lakukan proses <i>wrapping</i> dan <i>cropping</i> sesuai dengan <i>edge</i> objek utama dengan dimensi 256x256 piksel dengan format RGB sehingga diharapkan akan menghasilkan akurasi tinggi dan tentunya menghilangkan noise. Pada tahapan uji coba masing-masing dari tiap kelas telah di uji dengan tingkat akurasi tinggi, dan hasil akurasi keseluruhan model training tersebut adalah sebesar 97.1% akurasi.	Pada penelitian ini tidak di tampilkan hasil uji secara mendetail. akan lebih baik ditambahkan teknik-teknik augmentasi data atau segmentasi data untuk menghindari over/under fitting data. Sehingga berimbang pada akurasi dan validasi data	Pada penelitian ini, penulis menggunakan 2 skema pengujian, skema yang pertama adalah dengan menggunakan dataset asli, sedangkan yang kedua adalah dengan menggunakan dataset yang telah di lakukan dengan data augmentasi.

Tabel 2.1 (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
	Aplikasi Pendeteksi Penyakit Pada Daun Tanaman Apel dengan Metode CNN	-Guntur Wicaksono, Septi Andryana, Benrahma. -2020 - JOINTECS	Mendeteksi Penyakit Melalui Citra Daun Apel	Dalam penelitian ini arsitektur LeNet-5 digunakan dalam metode <i>Convolutional Neural Network</i> dengan mengolah 3151 dataset citra daun apel dengan tingkat akurasi 75%, peneliti disini menggunakan dataset yang berasal dari <i>Plang Village Crowd AI</i> . Penelitian ini menggunakan model arsitektur LeNet-5 dengan mengubah parameter agar meningkatkan hasil akurasi, yaitu menggunakan 32 maps yang berukuran 5 x 5 (C1) dengan maxpooling 2 x 2 (S1). Selanjutnya menggunakan 64 maps yang berukuran 5 x 5 (C2) dengan maxpooling 2 x 2 (S2). Kemudian dilanjutkan dengan memakai 2 hidden layer, hidden layer pertama memiliki 280 neuron dan pada hidden layer kedua memiliki 150 neuron. Kemudian untuk mengetahui tingkat akurasi model dengan menggunakan confusion matrix recall. Pengujian ini dilakukan dengan melakukan iterasi epoch yaitu epoch 50, epoch 75 dan 100 dimana epoch tertinggi memiliki tingkat akurasi tertinggi yaitu 97,8%. Hasil dari confusion matrix didapatkan hasil akhir akurasi model sebesar 89,62%	Pada penelitian ini dataset yang sudah cukup untuk membuat sebuah model, akan tetapi lebih bagus jika dataset yang digunakan ditambahkan. Kemudian untuk meningkatkan tingkat akurasi maka kita dapat menggunakan Teknik augmentasi data untuk meningkatkan akurasi data.	Pada penelitian mereka menggunakan metode <i>Convolutional Neural Network</i> dengan arsitektur LeNet. Sedangkan panulis menggunakan arsitektur <i>Convolutional Neural Network</i> dengan menambahkan teknik augmentasi data.

Tabel 2.1 (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
	Aplikasi Pendeteksi Penyakit Pada Daun Tanaman Apel dengan Metode CNN	-Guntur Wicaksono, Septi Andryana, Benrahma. - 2020 - JOINTECS	Mendeteksi Penyakit Melalui Citra Daun Apel	Dalam penelitian ini arsitektur LeNet-5 digunakan dalam metode <i>Convolutional Neural Network</i> dengan mengolah 3151 dataset citra daun apel dengan tingkat akurasi 75%, peneliti disini menggunakan dataset yang berasal dari <i>Plang Village Crowd AI</i> . Penelitian ini menggunakan model arsitektur LeNet-5 dengan mengubah parameter agar meningkatkan hasil akurasi, yaitu menggunakan 32 maps yang berukuran 5 x 5 (C1) dengan maxpooling 2 x 2 (S1). Selanjutnya menggunakan 64 maps yang berukuran 5 x 5 (C2) dengan maxpooling 2 x 2 (S2). Kemudian dilanjutkan dengan memakai 2 hidden layer, hidden layer pertama memiliki 280 neuron dan pada hidden layer kedua memiliki 150 neuron. Kemudian untuk mengetahui tingkat akurasi model dengan menggunakan confusion matrix recall. Pengujian ini dilakukan dengan melakukan iterasi epoch yaitu epoch 50, epoch 75 dan 100 dimana epoch tertinggi memiliki tingkat akurasi tertinggi yaitu 97,8%. Hasil dari confusion matrix didapatkan hasil akhir akurasi model sebesar 89,62%	Pada penelitian ini dataset yang sudah cukup untuk membuat sebuah model, akan tetapi lebih bagus jika dataset yang digunakan ditambahkan. Kemudian untuk meningkatkan tingkat akurasi maka kita dapat menggunakan Teknik augmentasi data untuk meningkatkan akurasi data.	Pada penelitian mereka menggunakan metode <i>Convolutional Neural Network</i> dengan arsitektur LeNet. Sedangkan panulis menggunakan arsitektur <i>Convolutional Neural Network</i> dengan menambahkan teknik augmentasi data.

Tabel 2.1 (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
4	Penggunaan Deep Learning Dengan Metode CNN untuk Klasifikasi Kualitas Sayur Kol berdasarkan Citra Fisik	- Ratna Dhamayanti, Mimin FathiyatuRohmah, Soffa Zahara - Jurnal Ilmiah Teknologi Informasi dan Sains (SUBMIT) - 2021	Mengklasifikasikan kualitas sayur kol berdasarkan citra dengan menggunakan metode CNN	Pada penelitian ini penulis menguji 270 dataset sayur kol yang di kelompokkan kedalam 2 label yaitu sayur kol yang baik/layak dan buruk serta untuk pembagiannya 210 data training dan 60 data testing. Pada penelitian ini penulis menggunakan skema dengan learning rate sebesar 0.004, iterasi epoch 30 dan 3 metode Stochastic Gradient Descent (SGD), Adaptive moment (Adam), dan Root Mean Square Propagation (RMSprop) dengan hasil akurasi tertinggi pada metode Adaptive Moment (Adam) dengan akurasi data testing 80 % dan 73% data training berdasarkan komposisi warna yang terdapat dalam citra.	Pada penelitian ini penulis hanya menggunakan satu fitur dalam mengklasifikasikan sayur kol yaitu dengan warna citra pada objek yang diteliti.	Pada penelitian ini lebih berfokus kepada algoritma performasi yaitu Algoritma <i>Stochastic Gradient Descent (SGD)</i> , <i>Adaptive Moment (Adam)</i> , dan <i>Root Mean Square Propagation (RMSProp)</i> sedangkan penulis berfokus kepada teknik <i>Data Augmentasi</i>

Tabel 2.1 (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
5	<i>Identification of Apple Leaf Diseases Based on Deep Convolutional Neural Networks</i>	-Bin Liu, Yun, Zhang, DongJian, He Yuxiang Li -Symetri -2018	Mengidentifikasi Penyakit Hama pada daun Apel dengan Deep Convolutional Neural Network	Pada penelitian ini terdapat 13.689 dataset daun apel yang telah di kelompokkan kedalam 4 jenis penyakit daun apel yaitu <i>Mosaic, Rust, Brown Spot</i> dan <i>Alternaria</i> digunakan untuk membuat model untuk mendeteksi penyakit pada daun apel, metode yang digunakan pada penelitian ini adalah dengan menggunakan metode Deep CNN. Hasil experiment menunjukkan bahwa model yang di buat pada penelitian ini mencapai tingkat akurasi tinggi yakni sebesar 97,62 % pada data testing dan tentunya lebih tinggi dari model tradisional lainnya, jika dibandingkan dengan model AlexNet standar, parameter yang di gunakan berkurang secara signifikan yaitu sebesar 51,206.928 dan menunjukan tingkat konvergensi yang lebih cepat. Kemudian metode yang dihasilkan terbukti lebih tinggi akurasinya yakni sebesar 10,83% jika menggunakan dataset sintesis dibanding dengan dataset asli.	Pada penelitian ini mereka menggunakan dataset private, sehingga untuk pengembangan lebih lanjut sulit untuk di lakukan oleh peneliti lain.	Penelitian ini menggunakan 13689 Citra sintentis daun apel yang telah di augmentasi dengan menggunakan teknik augmentasi <i>direction disturbance, ligh disturbance</i> dan <i>PCA Jittering</i> sedangkan penulis menggunakan teknik augmentasi dataset asli menggunakan teknik augmentasi <i>Random Rotation, horizontal flip</i> dan <i>vertical flip</i> .

Tabel 2.1 (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
6	<i>Disease Detection in Apple Leaves Using Image Processing Techniques</i>	Sara Alqethami, Badriah Almtanni, Walla Alzhani, Manal Alghamdi Engineering, Technology and Applied Science Research (ETASR), 2022	Penelitian ini bertujuan membuat sistem deteksi penyakit pada tanaman apel dengan menggunakan citra pada daun apel dengan menggunakan beberapa teknik image processing dan membandingkan 3 model prediksi yaitu <i>Convolutional Neural Network (CNN)</i> , <i>Support Vector Machine (SVM)</i>	Penelitian ini bertujuan membuat sistem deteksi hama pada daun apel menggunakan tiga metode prediksi model, diantaranya adalah dengan <i>Convolutional Neural Network (CNN)</i> , <i>Support Vector Machine (SVM)</i> dan <i>K-Nearest Neighbor (KNN)</i> pada penelitian ini menggunakan teknik image segmentation untuk meningkatkan tingkat akurasi pada 240 dataset yang di ujikan.		Penelitian ini menggunakan 3 model algoritma klasifikasi. Kemudian pada penelitian ini dilakukan teknik segmentasi data sedangkan penulis menggunakan algoritma <i>Convolutional Neural Network</i> data augmentasi.

Tabel 2.1 (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
7	Deteksi Penyakit pada Daun Kentang Menggunakan Pengolahan Citra dengan Metode Convolutional Neural Network	Abdul Jalil Rozaqi, Andi Sunyoto, Rudyanto Arief, Creative Information Technology (CITEC) Journal 2021	Deteksi Penyakit pada Daun kentang dengan menggunakan metode Convolutional Neural Network	Penelitian ini bertujuan untuk membuat sistem deteksi hama pada tanaman kentang melalui daun dengan menggunakan metode Convolutional Neural Network. Dataset public yang digunakan berjumlah 1152 yang telah di bagi kedalam 3 kelas dengan dimensi 150x150. Akurasi model pada pengujian ini adalah sebesar 95% pada data testing dan 94% pada data validasi.	Pada penelitian ini Jumlah dataset yang ada kurang proposional, dimana dataset antar kelas daun yang sehat cenderung lebih sedikit dari pada daun yang telah di labeley jenis hamanya.	Objek penelitian ini adalah daun kentang, kemudian pada penelitian ini dataset langsung di klasifikasikan dengan metode CNN sedangkan penulis dengan metode yang sama dilakukan dua skema pengujian dimana scenario pertama menggunakan dataset tanpa augmentasi dan skema yang kedua adalah dengan menggunakan data augmentasi.

2.3. Landasan Teori

2.3.1. Hama Tanaman

Hama dalam arti luas adalah semua bentuk gangguan baik pada manusia, ternak dan tanaman. Pengertian hama dalam arti yang lebih spesifik dan berkaitan dengan kegiatan budidaya tanaman adalah semua hewan yang merusak tanaman atau hasilnya yang mana aktivitas hidupnya ini dapat menimbulkan kerugian secara ekonomis. Adanya suatu hewan dalam satu pertanaman sebelum menimbulkan kerugian secara ekonomis maka dalam pengertian ini belum termasuk hama. Namun demikian potensi mereka sebagai hama nantinya perlu dimonitor dalam suatu kegiatan yang disebut pemantauan (*monitoring*). Secara garis besar hewan yang dapat menjadi hama dapat dari jenis serangga, moluska, tungau, tikus, burung, atau mamalia besar. Mungkin di suatu daerah hewan tersebut menjadi hama, namun di daerah lain belum tentu menjadi hama (Dadang, 2010).

Dalam penelitian ini terdapat 4 kelas pada daun apel yang telah diklasifikasikan dalam beberapa folder dataset, dari 4 kelas tersebut terdapat 3 jenis hama tersebut diantaranya adalah sebagai berikut:

1) Hama *Apple Scab*

Hama *Apple Scab* disebabkan oleh jamur *fungus venturia inaequalis*. Jenis jamur ini memiliki strain spesifik inang yang dapat menyebabkan penyakit pada satu jenis tanaman akan tetapi tidak untuk tanaman yang lain. Ciri-ciri dari hama ini adalah terdapat bintik-bintik bulat hitam pada daun, seiring bertambahnya ukuran daun bintik-bintik tersebut berubah menjadi coklat kehitam-hitaman dimana

hama tersebut tumbuh di sepanjang urat daun dan area dari sekitar bitnik tersebut akan mulai menguning. Buah yang terinfeksi hama ini akan memiliki kualitas yang buruk, bahkan cacat sebelum matang (Rebecca Koetter & Michelle Grabowski, 2019). Pada penelitian ini jumlah dataset hama ini adalah 741 gambar. Berikut gambar hama *Scab* pada daun apel:



Gambar 2.1. Hama *Scab* Pada Daun Apel

2) Hama *Apple Black Rot*

Hama *Apple Black Rot* disebabkan oleh jamur *Diplodia Seriata* (*syn Botryosphaeria obtusa*). Jamur ini dapat menginfeksi jaringan mati, serta jaringan hidup seperti batang, cabang, daun hingga buah. Jamur ini dapat menyebar dalam keadaan lembab. Spora dibawa oleh angin atau percikan air. Ciri-ciri dari daun apel yang terkena hama ini adalah area busuk berwarna coklat, pada buah biasanya akan membuat dagingnya mengkerut dan dalamnya biasanya busuk, biasanya bagian yang terkena hama ini akan mati dan lapuk karena terinfeksi jamur (Michelle Grabowski, 2019). Pada penelitian ini jumlah dataset hama ini adalah 733 gambar. Berikut gambar hama *black rot* pada daun apel:



Gambar 2.2. Hama *Black Rot* Pada Daun Apel

3) Hama *Apple Cedar/Rust*

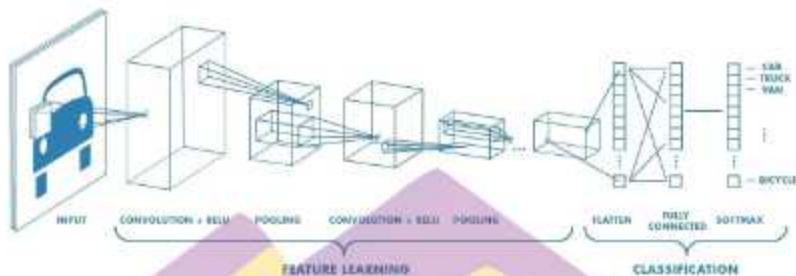
Hama *Apple Cedar/Rust* pada daun apel disebabkan oleh jamur *gymnosporangium* yang mempengaruhi Kesehatan dan kekuatan dari pohon apel tersebut karna dapat menyebabkan defoilisasi dini dan mengurangi kualitas buah. Pada kondisi terparahnya dapat menyebabkan pohon tersebut mati. Ciri-ciri utama dari hama ini adalah dengan adanya bitnik-bintik kuning kehijauan yang berangsur-angsur membesar pada daun disekelilingnya berwarna merah, pada bagian dalam terdapat bitnik-bintik kecil menyerupai jamur, jika diperbesar maka bitnik-bintik tersebut akan terlihat seperti buah berbulu (Jennifer Olson, 2017). Pada penelitian ini jumlah dataset hama ini adalah 447 gambar. Berikut contoh dari hama cedar/rust pada daun apel:



Gambar 2.3. Hama *Cedar Rust* Pada Daun Apel

2.3.2. Convolutional Neural Network

Metode *Convolutional Neural Network (CNN)* adalah salah satu kelas *deep-forward artificial neural networks* (Suyanto, 2018), dimana CNN ini terinspirasi dari proses-proses biologi pola konektivitas antar neurons menyerupai organisasi *visual cortex* pada binatang. *Cortical Neurons* menanggapi stimulasi hanya dalam suatu area terbatas pada bidang visual atau bidang reseptif (*reseptive field*). Bidang-bidang reseptif neurons tumpang tindih secara parsial (*partial overlap*) sedemikian hingga mencakup seluruh bidang reseptif (Zhang & Zhu, 2018). CNN terdiri dari lapisan masukan (*input layer*), lapisan luar (*output layer*) dan beberapa lapisan tersembunyi (*hidden layers*) *Hidden layers* terdiri dari *convolutional layers*, *pooling layers*, *normalization layers*, *ReLU Layer*, *fully connected layers* dan *loss layers*.



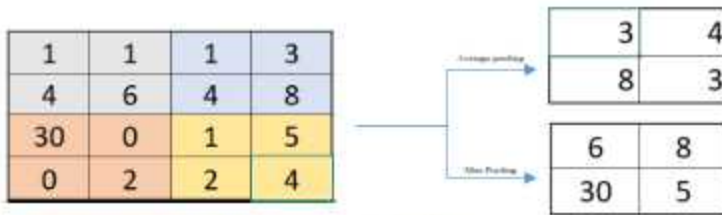
Gambar 2.4. Arsitektur *Convolutional Neural Network*.

2.3.2.1. *Convolutional Layers*

Convolutional Layer merupakan layer inti dari CNN dimana sebagian besar proses konvolusi dilakukan disini, Pada lapisan ini, operasi matematis konvolusi dilakukan antara input citra dan filter dengan ukuran tertentu, semisal $M \times M$, dengan menggeser filter diatas citra input, produk titik diambil dari bagian-bagian dari citra input yang terhubung dengan ukuran filter ($M \times M$).

2.3.2.2. *Pooling Layers*

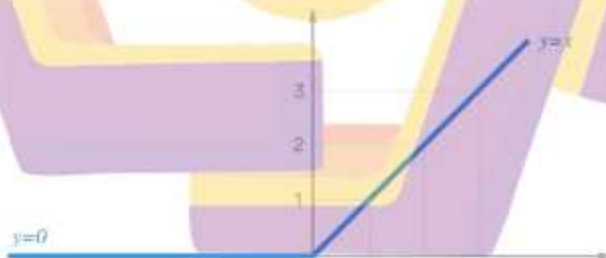
Pooling layers berfungsi menjaga ukuran data ketika proses convolution, yaitu dengan melakukan reduksi sample (*down sampling*) atau mereduksi input secara spasial (mengurangi jumlah parameter). Dengan *Pooling* kita dapat merepresentasikan data menjadi lebih kecil, mudah dikelola, dan mudah mengontrol *overfitting*. Kemudian dalam implementasinya tneknik pooling yang paling umum digunakan adalah *Max Pooling*, dimana dengan memilih nilai maksimum pada suatu area tertentu dan juga teknik lainnya adalah teknik *average pooling*, yakni dengan mencari nilai rata-rata dari area tertentu.



Gambar 2.5. Contoh Implementasi *Average Pooling* & *Max Pooling*

2.3.2.3. Rectified Linear Units Layer (ReLU Layer)

Rectified Linear Units (ReLU) layers berfungsi untuk menerapkan fungsi aktivasi $f(x) = x^+ = \max(0, x)$ nilai terhadap nilai output hasil konvolusi pada *convolutional layer*. Fungsi aktivasi ini meningkatkan sifat non linearitas fungsi keputusan dan jaringan secara keseluruhan tanpa mempengaruhi bidang-bidang reseptif pada *convolutional layer*. Berikut dibawah ini grafik fungsi aktivasi pada *ReLU Layer*.

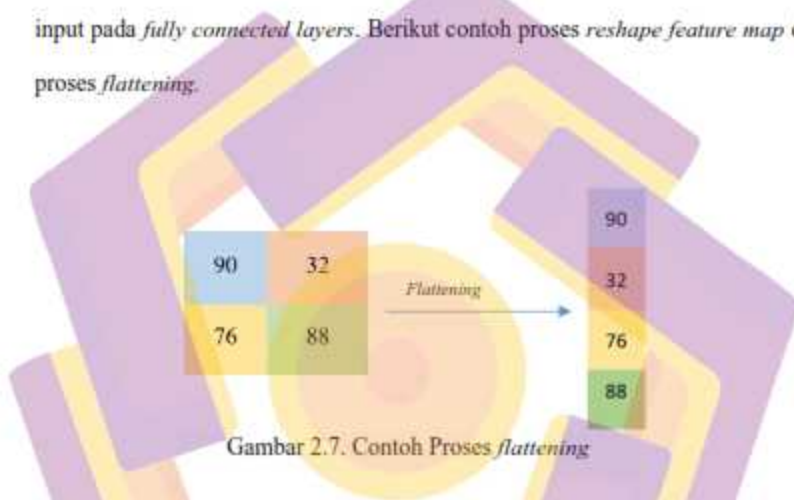


Gambar 2.6. Grafik fungsi aktivasi pada *ReLU Layer*.

2.3.2.4. Flattening

Tahapan *flattening* adalah tahapan merubah matriks yang ada di *pooling layer* menjadi satu kolom atau menjadi sebuah vektor tunggal, vektor ini akan

menjadi bagian dari input layer di *artificial neural network*, dari *pooled feature map* dari proses sebelumnya, kita cukup mengambil baris demi baris dan menggabungkannya menjadi satu baris 1 kolom. *Flatten* berfungsi untuk membentuk ulang (*reshape*) *feature map* dari *multimedimensional array* menjadi vector, hal tersebut dilakukan agar nilai-nilai tersebut dapat digunakan sebagai input pada *fully connected layers*. Berikut contoh proses *reshape feature map* dari proses *flattening*.

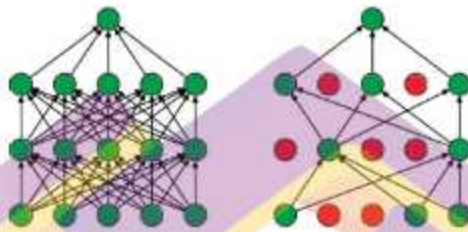


Gambar 2.7. Contoh Proses *flattening*

2.3.2.5 Dropout Regularization

Dropout Regulation adalah teknik regulasi *neural network* dimana beberapa *neuron* akan di pilih secara acak, dan tidak dipakai selama proses pelatihan data. dengan menghilangkan suatu *neuron* berarti menghilangkannya sementara dari *network* yang tidak menggunakan teknik *dropout*, dengan *neural network* yang menggunakan teknik *dropout*. Teknik ini akan membawa dampak pada performa model dalam melatih serta mengurangi *overfitting*. *Dropout* mengacu kepada menghilangkan neuron yang berupa *hidden layer* maupun *visible layer* di dalam

neuron. Setiap *neuron* akan diberikan probabilitas yang bernilai antara 0 dan 1. Berikut contoh penerapan *dropout regularization*.



Gambar 2.8. *Standard Network* (kiri) dan *Network dengan dropout regularization*

2.3.2.6. Fully Connected Layer

Pada lapisan fully connected layer setiap neurons memiliki koneksi penuh kesemua aktivasi dalam lapisan sebelumnya, hal ini sama persis dengan yang ada pada multilayer perception. Model aktivasinya yaitu dengan komputasi menggunakan suatu perkalian matriks yang diikuti dengan bias *offset*.

2.3.2.7. SoftMax

Dalam proses klasifikasi softmax menghitung probabilitas dari setiap kelas target terhadap semua kelas target yang ada. Rentang probabilitas output pada softmax adalah nilai dari 0 hingga 1.0, dan apabila semua nilai probabilitas dari kelas target tersebut dijumlahkan, maka nilainya akan sama dengan satu. Softmax menggunakan eksponensial dari nilai input yang diberikan dan jumlah nilai eksponensial dari semua nilai dalam output.

2.3.3. Augmentasi Data

Data Augmentasi adalah suatu proses dalam pengolahan data citra, augmentasi merupakan sebuah proses atau modifikasi gambar sedemikian rupa sehingga komputer akan mendeteksi bahwa gambar yang diubah adalah gambar yang berbeda, namun secara harfiah manusia masih mengenali gambar tersebut bahwa gambar tersebut adalah gambar yang sama. Augmentasi dapat meningkatkan akurasi model CNN yang sedang dibangun, karna augmentasi pada sebuah citra akan mendapatkan data-data baru yang membuat model melakukan generalisasi dengan lebih baik (Hasan Mahmud & al Faraby, 2019). Teknik augmentasi data dapat di bagi kedalam 2 kelompok kategori yaitu berdasarkan data lengkung (*data warping*) dan *data oversampling*. Berikut bagan pembagian kategori teknik augmentasi data (Amity University dkk., 2020).



Gambar 2.9. Pengelompokan Jenis Augmentasi Data

Berikut penjelasan dari bagan tersebut:

2.2.3.1. Augmentasi Data Berdasarkan Kelengkungan Data (*Data Warping*)

1) Transformasi Geometris (*Geometric Transformation*)

Salah satu teknik augmentasi data yang paling populer adalah dengan melakukan transformasi geometris pada sebuah citra. Teknik ini dibagi kedalam beberapa cara diantaranya adalah teknik *Flipping*, *Cropping*, *Rotation*, *Translation*, *noise injection*. Berikut contoh tranformasi geometris pada sebuah citra:



Gambar .2.10. Contoh augmentasi data

2) *Color Space Transformation*

Teknik ini melibatkan transformasi citra pertama dengan ruang warna multidimensi pertama RGB kedalam citra kedua dengan ruang warna multidimensi kedua CMYK. Gambar ditransformasikan menjadi 3 matriks bertumpuk. Ukuran setiap matriks direpresentasikan sebagai tinggi X lebar. Matriks menunjukkan nilai piksel untuk nilai warna RGB individual.

3) *Random Erasing*

Penghapusan acak (*random erasing*) adalah salah satu teknik data augmentasi dengan menghapus sebagian piksel secara kotak memanjang (*rectangle*) secara acak dan menghapus piksel didalam kotak memanjang tersebut dengan mengisi nilai dari setiap piksel yang ada pada kotak tersebut.(Zhong dkk., 2017)

4) *Adversarial Training*

Adversarial Training adalah salah satu teknik augmentasi data dengan memodifikasi citra asli dengan tujuan membuat neural network pada citra tersebut tersesat, dengan kata lain tujuannya dari adversarial ini adalah menambahkan *perturbation/reverse gradients* pada gambar yang asli dan membuatnya menghasilkan klasifikasi yang salah, berikut contoh penerapan *adversarial training*.



Gambar 2.11. Contoh Implementasi *Adversarial Training*

5) *Neural Style Transfer*

Neural Style Transfer adalah kemampuan sebuah *neural network* untuk mentransformasikan sebuah citra kedalam citra yang lainnya sehingga terbentuk

sebuah citra estetik dengan menggabungkan dua buah citra yang berbeda. Dua citra yang dimaksud adalah sebuah objek citra biasa dan sebuah objek citra *style* /lukis/abstrak(Jing dkk., 2019). Berikut contoh implementasi *neural style transfer*



Gambar 2.12. Contoh Penerapan *Neural Style Transfer*

2.2.3.2. Systemic Over

1) *Mixing Images/Sample Pairing*

Teknik ini *Mixing Images / Sample Pairing* adalah salah satu teknik augmentasi data dimana dua buah citra di gabungkan dan menghasilkan citra lainnya, kita dapat menghasilkan N^2 data baru dari N data *training* yang ada(Inoue, 2018). Teknik ini juga dapat ditingkatkan dengan membuat bentuk pencampuran gambar yang lebih umum, para peneliti juga menggunakan cara non-linear untuk mengintegrasikan gambar kedalam contoh data *training* baru, teknik pencampuran gambar ini juga bisa dilakukan dengan teknik lain seperti *random cropping* dan *patching*(Summers & Dinneen, 2019).

2) *Feature Space Augmentation*

Jaringan syaraf (*neural network*) sangat baik dalam memetakan (*mapping*) citra yang memiliki dimensi tinggi (*high-dimensional*) ke dimensi yang lebih rendah (*low dimensional*) pemetaan dapat dilakukan dengan kelas biner (*binary*

classes) maupun kedalam $n * 1$ vektor. Referensi tingkat rendah di lapisan tingkat yang lebih tinggi (*lower - level*) disebut *feature-space*. Menurut DeVries dan Taylor, noise dan ekstrapolasi adalah jenis augmentasi *feature-space*.(DeVries & Taylor, 2017)

3) *Generative Adversarial network*

Teknik augmentasi *Generative Adversarial Network (GAN)* digunakan untuk mengembangkan contoh data artifisial (*artificial sample*) dari dataset dan sample tersebut memiliki karakteristik seperti data aslinya(Bowles dkk., 2018). Teknik augmentasi *GAN* ini dapat menjadi salah satu cara untuk membuka informasi dari kumpulan data asli dengan menghasilkan sampel sintetesis atau data baru. Ada dua model *GAN* yaitu, *Generator Model* dan *Discriminator Model*. *Generator Model* akan membuat data baru dan *Discriminator Model* akan mengambil input dari sebuah domain (*real or generated*) dan memprediksi kelas biner yang dihasilkan itu asli atau palsu.

BAB III

METODE PENELITIAN

3.1. Jenis, Sifat, dan Pendekatan Penelitian

3.1.2. Jenis Penelitian

Jenis penelitian yang dilakukan oleh peneliti adalah penelitian kuantitatif dimana peneliti melakukan perhitungan matematis (*multidimensional matrix*) dimana output dari penelitian ini adalah berupa angka tingkat akurasi sebuah model yang dikembangkan.

3.1.2. Sifat Penelitian

Penelitian ini bersifat *experimental* dimana peneliti menerapkan beberapa teknik untuk meningkatkan akurasi pada model algoritma convolutional neural network pada ribuan dataset citra daun apel yang telah di beri label sesuai klasifikasinya.

3.1.3. Pendekatan Penelitian

Dalam penelitian ini peneliti menggunakan pendekatan penelitian kuantitatif dengan studi *experimental*, dimana peneliti melakukan *experiment* terhadap dataset yang ada untuk meningkatkan hasil akurasi model.

3.2. Metode Pengumpulan Data

Metode pengumpulan yang dilakukan peneliti adalah dengan melakukan studi literatur dan dokumen. Data yang terkumpul bersumber dari repositori dataset Kaggle dimana dataset yang di uji adalah public dataset. Pada penelitian ini peneliti mengumpulkan ribuan dataset citra daun apel yang telah di beri label sesuai dengan klasifikasi penyakitnya.

3.3. Metode Analisis Data

Pada bagian ini peneliti menggunakan aplikasi jupyter notebook dengan framework tensor flow dan keras serta beberapa library pendukung untuk membangun sebuah neural network. Dataset ini berupa citra daun apel berukuran 256x256 piksel yang telah dikumpulkan dan telah dilabeli di pisahkan sesuai dengan klasifikasinya, kemudian dataset yang ada di olah dengan menggunakan jupyter notebook dimana pada proses ini peneliti melakukan beberapa skenario dimana skenario pertama peneliti akan melakukan teknik data augmentation untuk meningkatkan jumlah dataset yang ada, teknik augmentasi yang dilakukan adalah dengan teknik vertical flip, horizontal flip dan teknik rotasi citra. Kemudian skenario kedua peneliti akan melakukan training langsung dengan model yang akan dibangun dengan data yang ada.

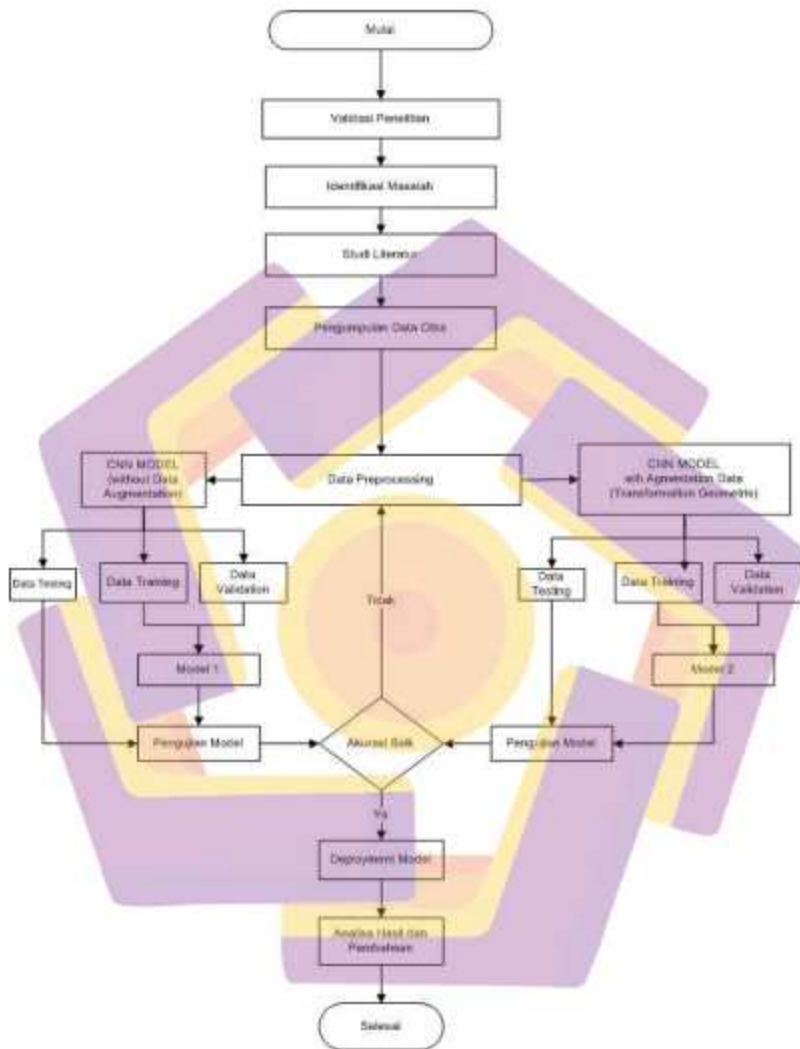
Pada skenario pertama setelah dilakukan augmentasi maka langkah selanjutnya adalah melakukan pembagian data *training* sebanyak 80% data testing sebanyak 10% dan data *validation* sebanyak 10% pada tahap ini peneliti membangun model dengan arsitektur convolutional neural network. Untuk

meningkatkan akurasi model peneliti menggunakan algoritma Adam Optimizer, yaitu algoritma optimisasi yang dapat digunakan sebagai ganti dari prosedur classical stochastic gradient descent untuk memperbarui bobot secara iteratif yang didasarkan pada data training. Untuk skenario kedua peneliti akan melakukan hal yang sama pada data yang ada untuk dilakukan training data akan tetapi tanpa metode augmentasi data.

Setelah tahap pembuatan model maka langkah selanjutnya adalah dengan melakukan pengujian model yang telah di bangun,, dimana pada tahap ini kita akan menguji dua model yang telah kita bangun mana yang akan memiliki tingkat akurasi tertinggi.

3.4. Alur Penelitian

Pada penitian ini kita dapat melihat flowchart yang penulis rancang berdasarkan alur penelitian yang berjalan, berikut gambar flowchart penelitian dibawah ini:



Gambar 3.1. Flowchart Penelitian

BAB IV

HASIL PENELITIAN DAN PEMBAHASAN

4.1. Analisis Kebutuhan Data

Tahap analisis kebutuhan data merupakan salah satu proses penting dalam penelitian ini, kebutuhan data yang dimaksud adalah kebutuhan data yang akan digunakan dalam proses *training* model *machine learning* yang akan dibangun. Pada penelitian ini data yang dibutuhkan adalah dataset citra dan apel yang telah di kelompokkan sesuai dengan jenis hama yang telah teridentifikasi.

4.2. Data Citra Daun Apel

Dataset Citra daun apel yang didapatkan oleh penulis adalah hasil eksplorasi dataset publik yang diperoleh dari beberapa situs penyedia *dataset public* salah satunya adalah [Kaggle.com](https://www.kaggle.com) terdiri dari empat kelompok jenis daun apel yang telah diberi label sesuai dengan hama yang ada pada daun tersebut dengan total dataset sebanyak 3760 jenis daun apel yang di bagi kedalam empat kelompok. Keempat jenis citra daun tersebut adalah citra daun apel sehat (*healthy*) berjumlah 1839 data, citra daun *Apple Scab* yang disebabkan oleh oleh jamur *venturia inaequalis* berjumlah 741 data, citra daun *Apple Black Rot* oleh jamur *Botryosphaeria Obtusa* berjumlah 733 data dan citra *Apple Cedar /Apple Rust* yang disebabkan oleh jamur *Gymnosporangium Juniperi-Virginianae* sebanyak 447 data. Berikut contoh dari citra dataset daun apel pada Tabel 4.1. dibawah ini.



Gambar 4.1. Daun Apel Sehat (a), hama daun *Apple Scab* (b), hama daun *Apple Black Rot* (c), hama daun *Apple Cedar* (d)

4.3. Pengujian Metode

Pada tahap ini penulis akan melakukan beberapa skenario pengujian model diantaranya adalah skenario pengujian dengan dataset asli dan skenario pengujian dengan menerapkan teknik augmentasi data. Adapun teknik-teknik augmentasi data yang diterapkan pada penelitian ini adalah teknik augmentasi data transformasi geometris, diantaranya adalah teknik *horizontal flip*, *vertical flip*, *random rotation* dan *Random brightness*.

Dalam prosesnya setiap skenario akan ditentukan inputan parameter-parameter modelnya yang ditulis dalam bahasa pemrograman *python* dengan tools *Jupyter notebook* serta *library-library machine learning* lainnya. Kemudian penulis melakukan inisiasi beberapa parameter penting diantaranya adalah jumlah *channels* 3, *batch size* 32 dengan jumlah *epoch* 60. Setiap parameter yang ada di terapkan kedalam dataset yang telah dibagi kedalam 80% *dataset training*, 10% *data testing* dan 10% *data validation*. Pada penelitian ini juga penulis melakukan *image rescaling/resizing* menjadi 256x256 dengan menggunakan *library image data generator* dari *library tensor flow python* piksel untuk seluruh dataset yang ada

untuk mengoptimasikan kecepatan proses komputasi citra (Amity University & Institute of Electrical and Electronics Engineers, 2020). Semua input konfigurasi tersebut akan diimplementasikan kedalam model CNN yang akan dibangun. Untuk model yang telah dibangun akan dioptimasi dengan algoritma *Adaptive Moment Estimation (Adam Optimizer)*(Kingma & Ba, 2014), ADAM adalah algoritma optimisasi pengganti untuk *stochastic gradient descent* untuk *training model* dalam *deep learning* yang menggabungkan sifat-sifat terbaik dari algoritma *AdaGrad* dan *RMSProp* untuk memberikan algoritma yang lebih optimal dimana dapat menangani gradien yang menyebar dan memiliki *noise*. Berikut listing source code dan inialisasi parameter-parameter pada model yang akan dibangun:

No	Source Code
1	<pre>import tensorflow as tf from tensorflow.keras import models, layers import matplotlib.pyplot as plt from IPython.display import HTML from tensorflow.keras.preprocessing.image import ImageDataGenerator import splitfolders</pre>
2	<pre>splitfolders.ratio("apple_dataset", output="dataset2", seed=1337, ratio=(.8, .1, .1), group_prefix=None, move=False) # default values splitfolders.fixed("apple_dataset", output="dataset2",</pre>

	<pre>seed=1337, fixed=(100, 100), oversample=False, group_prefix=None, move=False) # default values</pre>
3	<pre>IMAGE_SIZE = 256 CHANNELS=3 BATCH_SIZE=32 EPOCHS = 60</pre>
4	<pre>train_datagen =ImageDataGenerator(rescale=1./255, horizontal_flip=True, rotation_range=20, vertical_flip=True,) train_generator = train_datagen.flow_from_directory('dataset2/train', target_size =(IMAGE_SIZE,IMAGE_SIZE), batch_size = 32, class_mode = 'sparse')</pre>
5	<pre>validation_datagen =ImageDataGenerator(rescale=1./255, horizontal_flip=True, rotation_range=20, vertical_flip=True,) validation_generator = validation_datagen.flow_from_directory('dataset2/val', target_size =(IMAGE_SIZE,IMAGE_SIZE),</pre>

	<pre>batch_size = 32, class_mode = 'sparse')</pre>
--	--

Listing 4.1. *Source Code* Tahapan Inisiasi Model CNN

Berikut penjelasan kode program pada listing 4.1:

- a) *Cell 1* Import library yang di butuhkan yang ada pada python
- b) *Cell 2 Split Folder Training*. membagi dataset kedalam 3 bagian yaitu data training sebanyak 80%, data testing 10% dan data validation 10%
- c) *Cell 3* Inisialisai parameter ukuran gambar, jumlah *channel*, *batch size* dan *epoch*
- d) *Cell 4* Membuat fungsi *train data generator* untuk *data training* dengan merubah ukuran citra menjadi 256x255, dan menentukan jenis augmentasi yang akan dilakukan.
- e) *Cell 5* membuat fungsi *validationa data generator* untuk validasi data dengan merubah citra menjadi 256x256 dan menentukan jenis augmentasi yang akan dilakukan.

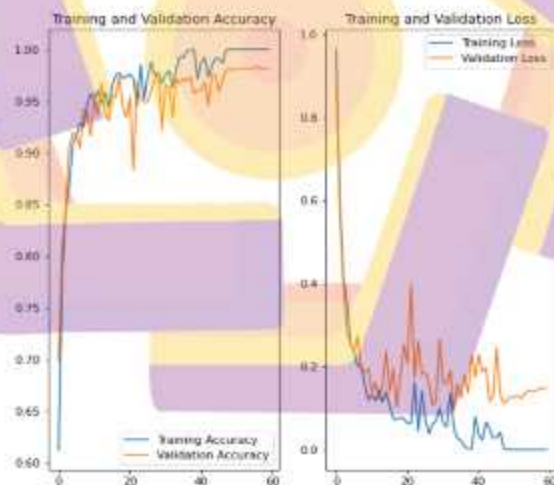
4.3.1 Pengujian Metode CNN dengan Dataset Asli

Pada skenario pertama ini penulis membangun model dengan dataset asli, dimana tahap pertama adalah dengan melakukan inisiasi parameter seperti pada pembahasan diatas, kemudian dataset yang digunakan pada skenario pertama ini adalah sebagai berikut:

Table 4.1. Sebaran Dataset Asli

Jenis Kelas Daun	Data Training (80%)	Data Validation (10%)	Data Testing (10%)	Total
<i>Apple Scab</i>	504	137	100	741
<i>Apple Black Root</i>	496	137	100	733
<i>Apple Cedar/rust</i>	220	127	100	447
<i>Apple Healthy</i>	1445	229	165	1839
Total	2665	630	465	3760

Hasil evaluasi model pada skenario pertama ini memiliki *validation accuracy* akurasi sebesar 0.9398 (93.98%) dengan nilai *loss validation* 0.04286 (4.2%). Berikut Grafik *training accuracy* dan *validation accuracy*:

Gambar 4.2. Grafik *accuracy* & *loss* CNN dengan dataset asli

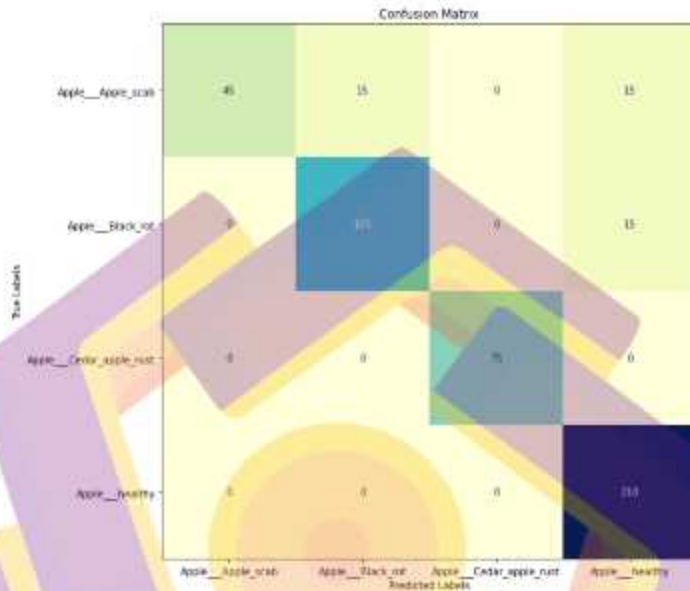
Berdasarkan grafik diatas bisa kita lihat terlihat terjadi *overfitting* pada model yang di bangun dimana semakin tinggi *epoch*, nilai *validation loss* semakin tinggi, artinya model yang dibangun kurang baik dan akan menciptakan model yang

buruk. Berikut penulis sajikan contoh hasil prediksi model dengan dataset asli dalam tabel dibawah ini:

Table 4.2. Tabel Prediksi Model CNN dataset asli

No	Data Aktual (<i>Actual class</i>)	Prediksi Kelas (<i>predicted class</i>)	Nilai Kepercayaan (<i>confidence</i>)
1	<i>Apple Scab</i>	<i>Apple Scab</i>	100%
2	<i>Apple Healthy</i>	<i>Apple Healthy</i>	100%
3	<i>Apple Healthy</i>	<i>Apple Healthy</i>	99.98%
4	<i>Apple Black Rot</i>	<i>Apple Healthy</i>	99.5%
5	<i>Apple Black Rot</i>	<i>Apple Black Rot</i>	98.8%
6	<i>Apple Scab</i>	<i>Apple Healthy</i>	99.77%
7	<i>Apple Healthy</i>	<i>Apple Healthy</i>	100%
8	<i>Apple Black Rot</i>	<i>Apple Black Rot</i>	100%
9	<i>Apple Healthy</i>	<i>Apple Healthy</i>	100%

Dari tabel diatas kita dapat melihat ada beberapa dataset yang bernilai *false negative*. Lebih detailnya penulis sajikan dalam *plot confusion matrix* dibawah ini:



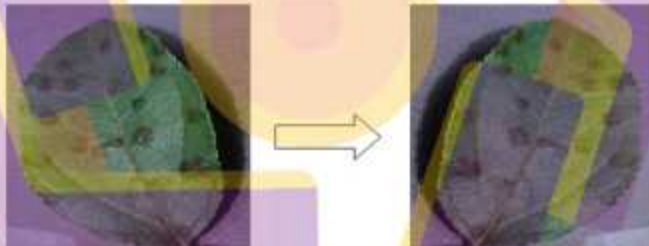
Gambar 4.3. Confusion Matrix Model CNN Dengan Dataset Asli

Berdasarkan *plot confusion matrix* diatas kita dapat melihat pada kelas *apple scab* dan *apple black rot* terdapat nilai *false negative*. Pada pengujian model ini total *train accuracy* memiliki *score 100%*, total *test accuracy 90.62%*, total *precision score 90.62%* dan total *recall score 90.62%*.

4.3.2 Pengujian Metode CNN dengan Teknik Augmentasi Data

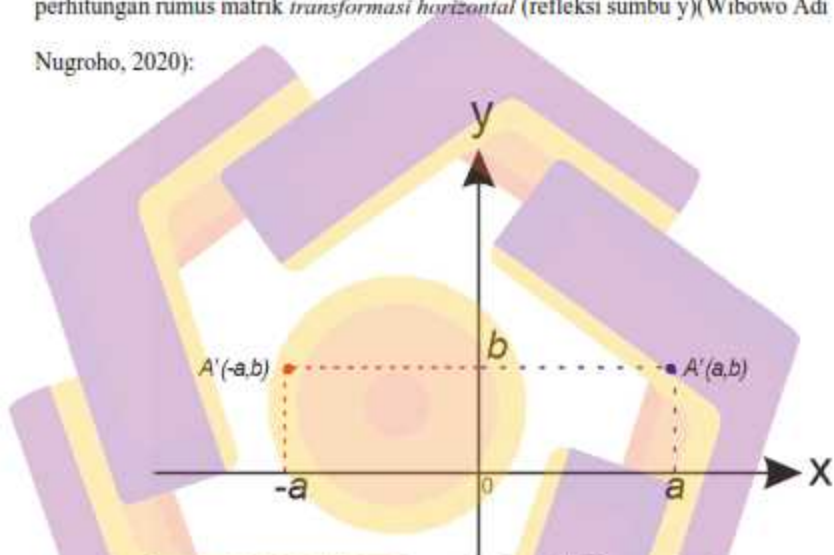
a. Pengujian Model CNN Augmentasi Data dengan Teknik *Horizontal Flip*

Pada skenario pengujian dengan data augmentasi pertama, penulis melakukan pengujian model CNN dengan teknik augmentasi data transformasi geometris horizontal flip. Teknik ini adalah teknik mentransformasikan sebuah citra secara horizontal, dengan kata lain setiap citra asli akan mempunyai duplikasi citra baru hasil transformasi geometris secara horizontal yang di generate secara realtime pada setiap dataset yang di training dengan menggunakan library image preprocessing tensorflow and keras sehingga total dataset akan bertambah dua kali lipat dari data asli. Berikut contoh transformasi citra secara *horizontal flip* (pencerminan).



Gambar 4.4. Contoh transformasi citra secara *horizontal (horizontal flip)*

Berdasarkan dari contoh diatas kita dapat melihat gambar sebelah kiri adalah dataset asli dan yang sebelah kanan adalah dataset hasil dari transformasi geometris secara refleksi/flip, sehingga setiap dataset akan mengenerate data baru untuk ditambahkan dalam setiap *data training*. untuk lebih jelasnya berikut perhitungan rumus matrik *transformasi horizontal* (refleksi sumbu y)(Wibowo Adi Nugroho, 2020):



Gambar 4.5. Matrik Transformasi horizontal (Pencerminan)

Berikut Matriks Transformasi dari gambar pencerminan diatas:

$$A(a, b) \rightarrow A' = \begin{pmatrix} a' \\ b' \end{pmatrix} = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} -a \\ b \end{pmatrix} \quad (4.1)$$

Keterangan Rumus:

- A adalah titik A
- A' adalah hasil pencerminan dari titik A
- a adalah titik yang ada pada sumbu x
- b adalah titik yang ada pada sumbu y

Contoh Transformasi refleksi *horizontal* (sumbu y) pada matriks 2 dimensi. Jika diketahui titik B (2,5) di refleksikan terhadap sumbu x maka bayangan titik B adalah sebagai berikut :

$$B(2,5) \longrightarrow B'(a',b')$$

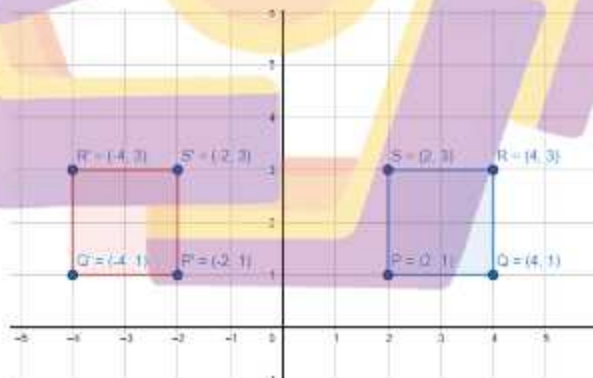
$$\begin{pmatrix} a' \\ b' \end{pmatrix} = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 2 \\ 5 \end{pmatrix} = \begin{pmatrix} -2 \\ 5 \end{pmatrix}$$

Maka bayangan titik B adalah $B'(-2,5)$.

Dalam contoh lain Jika contoh gambar dataset diatas di ilustrasikan kedalam matrik 2 dimensi maka bisa di ilustrasikan sebagai 4 titik koordinat P(2,1), Q(4,1),R(4,3),S(2,3) dalam sebuah matriks sebagai berikut:

$$\begin{pmatrix} a' \\ b' \end{pmatrix} = \begin{pmatrix} (2,1) & (4,1) \\ (4,3) & (2,3) \end{pmatrix} \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} (-2,1) & (-4,1) \\ (-4,3) & (-2,3) \end{pmatrix}$$

Berikut ilustrasi dalam matriks dari persamaan diatas:



Gambar 4.6. Ilustrasi Matrik Transformasi refleksi *horizontal*

Dalam pengujian ini berikut potongan *source code* yang digunakan untuk melakukan training data dengan implementasi teknik augmentasi data *horizontal flip* pada model CNN:

No	Source Code
1	<pre> IMAGE_SIZE = 256 CHANNELS=3 BATCH_SIZE=32 EPOCHS = 60 train_datagen =ImageDataGenerator(#Image Resizing menjadi 256x256 rescale=1./255, horizontal_flip=True) train_generator = train_datagen.flow_from_directory('dataset2/train', target_size =(IMAGE_SIZE, IMAGE_SIZE), batch_size = 32, class_mode = 'sparse') </pre>
2	<pre> validation_datagen =ImageDataGenerator(rescale=1./255, horizontal_flip=True) validation_generator= validation_datagen.flow_from_directory('dataset2/val', target_size =(IMAGE_SIZE, IMAGE_SIZE), batch_size = 32, class_mode = 'sparse') </pre>
3	<pre> test_datagen =ImageDataGenerator(rescale=1./255, horizontal_flip=True) test_generator = test_datagen.flow_from_directory('dataset2/test', target_size =(IMAGE_SIZE, IMAGE_SIZE), batch_size = 32, class_mode = 'sparse') </pre>
4	<pre> input_shape = (IMAGE_SIZE, IMAGE_SIZE, CHANNELS) n_classes =4 </pre>

	<pre> model = models.Sequential([layers.InputLayer(input_shape = input_shape), layers.Conv2D(32, kernel_size=(3, 3), activation='relu'), layers.MaxPooling2D((2, 2)), layers.Conv2D(64, kernel_size=(3, 3), activation='relu'), layers.MaxPooling2D((2, 2)), layers.Conv2D(64, kernel_size=(3, 3), activation='relu'), layers.MaxPooling2D((2, 2)), layers.Conv2D(64, (3, 3), activation='relu'), layers.MaxPooling2D((2, 2)), layers.Conv2D(64, (3, 3), activation='relu'), layers.MaxPooling2D((2, 2)), layers.Flatten(), layers.Dense(64, activation='relu'), layers.Dense(n_classes, activation='softmax'),]) </pre>
5	<pre> model.compile(optimizer='adam', loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=False), metrics=['accuracy']) </pre>

Listing 4.2. Source Code Training Model CNN dengan Augmentasi Data

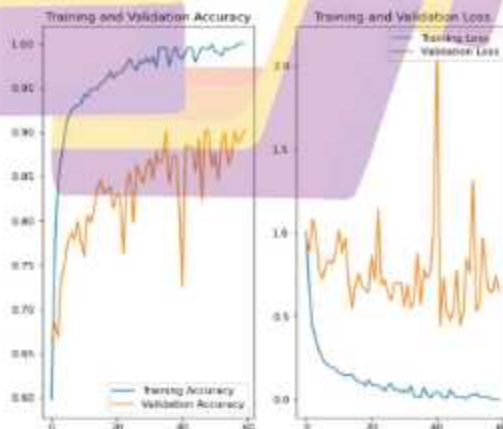
Horizontal Flip

Keterangan kode program diatas:

- a) Cell 1, 4 Baris pertama adalah konfigurasi parameter dasar meliputi *Image size*, *channels*, *batchsize* dan *epoch*. Pada *cell* ini juga terdapat fungsi untuk membuat *generator data* augmentasi secara *horizontal flip* dan *me-rescale* gambar menjadi 255x255. Pada bagian *train_generator* adalah sebuah fungsi untuk mengatur sumber dataset yang akan diaugmentasikan.

- b) *Cell 2*, berisi fungsi untuk menggenerate *data validation* dengan augmentasi *horizontal flip* dan *rescaling* gambar menjadi 256x256 piksel. Kemudian terdapat fungsi untuk menentukan sumber *dataset validation* yang akan di gunakan.
- c) *Cell 3*, fungsi untuk menggenerate *data testing* dengan augmentasi *horizontal flip* dan *rescaling* gambar menjadi 256x256 piksel. Kemudian terdapat fungsi untuk menentukan sumber *dataset testing* yang akan di gunakan.
- d) *Cell 4*, baris kode ini adalah input model CNN dengan parameter-parameter yang telah ditentukan di diawal
- e) *Cell 5*, adalah baris kode untuk mengoptimasi model yang akan di training dengan algoritma *Adam Optimizer*.

Hasil evaluasi model pada skenario pertama ini memiliki *validation accuracy* akurasi sebesar 0.8838 (88.38%) dengan nilai *loss validation* 0.5728 (5.7%). Berikut grafik *training accuracy* dan *validation loss* pada skenario pertama ini:



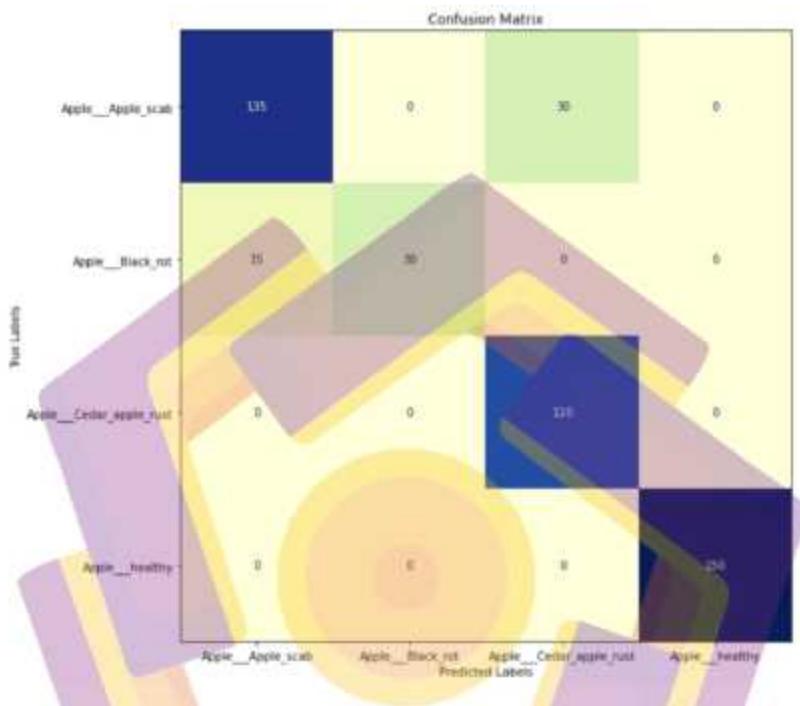
Gambar 4.7. Grafik *accuracy* & *loss* model CNN teknik augmentasi *horizontal flip*.

Grafik diatas menunjukkan pergerakan nilai dari accuracy & loss dari data training dan data validation, dimana pada grafik pertama nilai dari training accuracy dan validation accuracy berbeda cukup signifikan, begitupun pada grafik training dan validation loss hal ini menunjukkan adanya overfitting pada model yang dibangun. Berikut hasil test prediksi pada dataset testing dari model tersebut.

Tabel 4.3 Prediksi Model CNN dengan teknik augmentasi data *horizontal flip*.

No	Data Aktual (<i>Actual class</i>)	Prediksi Kelas (<i>predicted class</i>)	Nilai Kepercayaan (<i>confidence</i>)
1	<i>Apple Cedar/Rust</i>	<i>Apple Cedar/Rust</i>	99.99%
2	<i>Apple Black Rot</i>	<i>Apple Black Rot</i>	100%
3	<i>Apple Cedar/Rust</i>	<i>Apple Cedar/Rust</i>	100%
4	<i>Apple Healthy</i>	<i>Apple Healthy</i>	100%
5	<i>Apple Scab</i>	<i>Apple Scab</i>	100%
6	<i>Apple Cedar/Rust</i>	<i>Apple Cedar/Rust</i>	100%
7	<i>Apple Black Rot</i>	<i>Apple Cedar/Rust</i>	99.47%
8	<i>Apple Black Rot</i>	<i>Apple Scab</i>	98.89%
9	<i>Apple Cedar/Rust</i>	<i>Apple Cedar/Rust</i>	100%

berikut hasilnya penulis gambarkan dalam sebuah plot *confusion matrix*:

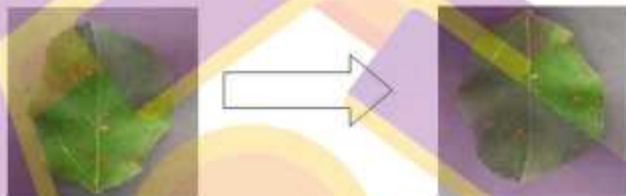


Gambar 4.8. *Confusion Matrix CNN* dengan augmentasi data horizontal flip

Dari gambar diatas kita dapat melihat bahwa terdapat beberapa kesalahan prediksi dari 2 kelas yang ada dengan nilai perhitungan *test accuracy 100%*, *precision score 90.62%*, *total recall score 90.62%* dan *total recall 90.62%*.

b. Pengujian Model CNN Augmentasi Data dengan Teknik Vertical Flip

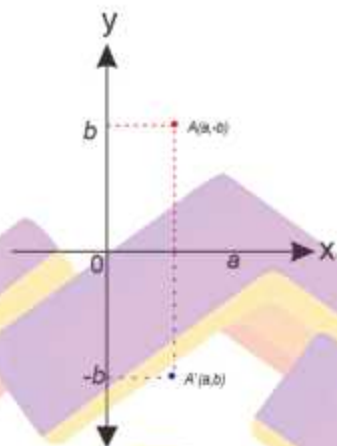
Pengujian skenario kedua, penulis melakukan pengujian model CNN dengan teknik augmentasi data transformasi geometris *vertical flip*. Teknik ini adalah teknik mentransformasikan sebuah citra secara *vertical*, dengan kata lain data yang baru akan di *generate* seperti cerminan dari atas kebawah atau sebaliknya dan total dataset akan bertambah dua kali lipat dari dataset asli. Berikut contoh dataset transformasi *vertical flip*:



Gambar 4.9. Contoh transformasi citra secara *vertical (vertical flip)*

Berdasarkan dari contoh diatas kita dapat melihat gambar sebelah kiri adalah dataset asli dan yang sebelah kanan adalah dataset hasil dari transformasi geometris secara refleksi/flip secara *vertical*, sehingga setiap dataset akan mengenerate data baru untuk ditambahkan dalam setiap *data training*.

untuk lebih jelasnya berikut perhitungan rumus matrik transformasi *vertical (refleksi sumbu x)* (Wibowo Adi Nugroho, 2020):



Gambar 4.10. Matriks transformasi *vertical* (*vertical flip*)

Berikut Matriks Transformasi dari gambar pencerminan diatas:

$$A(a, b) \rightarrow A' = \begin{pmatrix} a' \\ b' \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} a \\ -b \end{pmatrix} \quad (4.1)$$

Keterangan Rumus:

- A adalah titik A
- A' adalah hasil pencerminan dari titik A
- a adalah titik yang ada pada sumbu x
- b adalah titik yang ada pada sumbu y

Berikut contoh transformasi matriks 2 dimensi pada sumbu x (*vertical flip*):

Jika diketahui titik B (2,5) di refleksikan terhadap sumbu x maka bayangan titik B adalah sebagai berikut :

$$B(2,5) \longrightarrow B'(a', b')$$

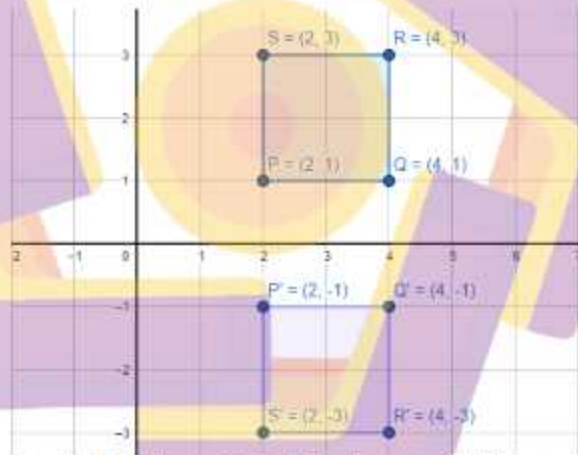
$$\begin{pmatrix} a' \\ b' \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} 2 \\ 5 \end{pmatrix} = \begin{pmatrix} 2 \\ -5 \end{pmatrix}$$

Maka bayangan titik B adalah $B'(2,-5)$.

Dalam contoh lain Jika contoh gambar dataset diatas di ilustrasikan kedalam matrik 2 dimensi maka bisa di ilustrasikan sebagai 4 titik koordinat $P(2,1)$, $Q(4,1)$, $R(4,3)$, $S(2,3)$ dalam sebuah matriks sebagai berikut:

$$\begin{pmatrix} a' \\ b' \end{pmatrix} = \begin{pmatrix} (2,1) & (4,1) \\ (4,3) & (2,3) \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} = \begin{pmatrix} (2,-1) & (4,-1) \\ (4,-3) & (2,-3) \end{pmatrix}$$

Berikut ilustrasi dalam matriks dari persamaan diatas:



Gambar 4.11. Ilustrasi Matrik Transformasi refleksi vertical

Pada penelitian ini penulis menggunakan *tools jupyter notebook* dengan bahasa pemrograman *python* untuk membuat modelnya, berikut potongan *source code* yang digunakan untuk melakukan training data dengan implementasi teknik augmentasi data *vertical flip* pada model CNN:

No	Source Code
1	<pre> IMAGE_SIZE = 256 CHANNELS=3 BATCH_SIZE=32 EPOCHS = 60 train_datagen =ImageDataGenerator(#Image Resizing menjadi 256x256 rescale=1./255, vertical_flip=True) train_generator = train_datagen.flow_from_directory('dataset2/train', target_size =(IMAGE_SIZE, IMAGE_SIZE), batch_size = 32, class_mode = 'sparse') </pre>
2	<pre> validation_datagen =ImageDataGenerator(rescale=1./255, vertical_flip=True) validation_generator= validation_datagen.flow_from_directory('dataset2/va l', target_size =(IMAGE_SIZE, IMAGE_SIZE), batch_size = 32, class_mode = 'sparse') </pre>
3	<pre> test_datagen =ImageDataGenerator(rescale=1./255, vertical_flip=True) test_generator = test_datagen.flow_from_directory('dataset2/test', target_size =(IMAGE_SIZE, IMAGE_SIZE), batch_size = 32, class_mode = 'sparse') </pre>
4	<pre> input_shape = (IMAGE_SIZE, IMAGE_SIZE, CHANNELS) n_classes =4 model = models.Sequential([layers.InputLayer(input_shape = input_shape), layers.Conv2D(32, kernel_size=(3, 3), activation='relu'), layers.MaxPooling2D((2, 2)), layers.Conv2D(64, kernel_size=(3, 3), activation='relu'), layers.MaxPooling2D((2, 2)), </pre>

	<pre> layers.Conv2D(64, kernel_size=(3, 3), activation='relu'), layers.MaxPooling2D((2, 2)), layers.Conv2D(64, (3, 3), activation='relu'), layers.MaxPooling2D((2, 2)), layers.Conv2D(64, (3, 3), activation='relu'), layers.MaxPooling2D((2, 2)), layers.Conv2D(64, (3, 3), activation='relu'), layers.MaxPooling2D((2, 2)), layers.Flatten(), layers.Dense(64, activation='relu'), layers.Dense(n_classes, activation='softmax'),)) </pre>
5	<pre> model.compile(optimizer='adam', loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=False), metrics=['accuracy']) </pre>

Listing 4.3. *Source Code Training Model CNN dengan Augmentasi Data vertical*

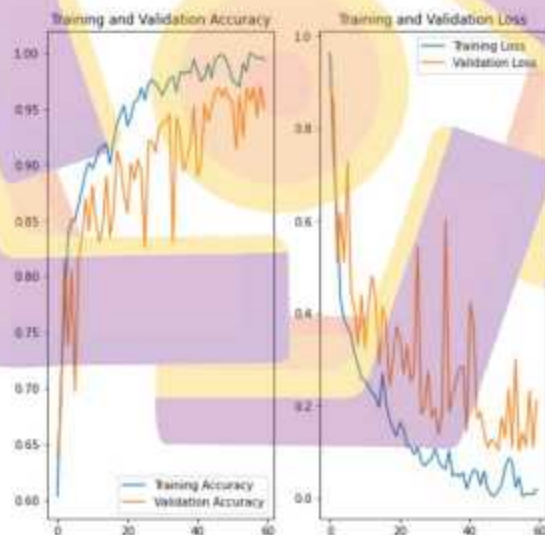
Flip

Keterangan kode program diatas:

- Cell 1*, 4 Baris pertama adalah konfigurasi parameter dasar meliputi *Image size*, *channels*, *batchsize* dan *epoch*. Pada *cell* ini juga terdapat fungsi untuk membuat *generator data* augmentasi secara *vertical flip* dan *re-scale* gambar menjadi 255x255. Pada bagian *train_generator* adalah sebuah fungsi untuk mengatur sumber dataset yang akan diaugmentasikan.
- Cell 2*, berisi fungsi untuk menggenerate *data validation* dengan augmentasi *vertical flip* dan *rescaling* gambar menjadi 256x256 piksel. Kemudian terdapat fungsi untuk menentukan sumber *dataset validation* yang akan di gunakan.

- c) *Cell 3*, fungsi untuk menggenerate *data testing* dengan augmentasi *vertical flip* dan *rescaling* gambar menjadi 256x256 piksel. Kemudian terdapat fungsi untuk menentukan sumber *dataset testing* yang akan di gunakan.
- d) *Cell 4*, baris kode ini adalah input model CNN dengan parameter-parameter yang telah ditentukan di awal
- e) *Cell 5*, adalah baris kode untuk mengoptimasi model yang akan di training dengan algoritma *Adam Optimizer*.

Hasil evaluasi model pada skenario kedua ini memiliki *validation accuracy* akurasi sebesar 0.933 (93.3%) dengan nilai *loss validation* 0.2125(21.25%).Berikut grafik *Training accuracy* dan *Validation Loss* pada skenario kedua ini:



Gambar 4.12. Grafik accuracy & loss model CNN teknik augmentasi *vertical flip*.

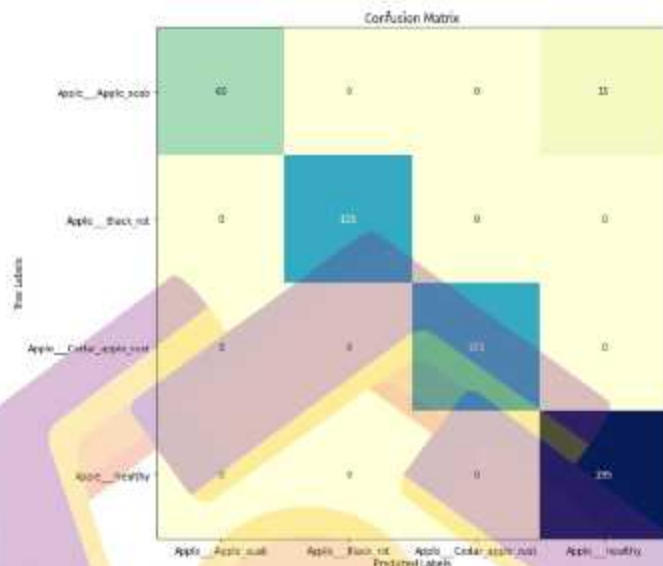
Grafik diatas menunjukkan pergerakan nilai dari *accuracy* & *loss* dari *data training* dan *data validation*, dimana pada grafik pertama nilai dari *training*

accuracy dan *validation accuracy* berbeda cukup signifikan, begitupun pada grafik *training* dan *validation loss* hal ini menunjukkan adanya *overfitting* pada model yang dibangun. Berikut hasil test prediksi pada *dataset testing* dari model tersebut.

Tabel 4.4. Prediksi Model CNN dengan teknik augmentasi data *vertical flip*.

No	Data Aktual (<i>Actual class</i>)	Prediksi Kelas (<i>predicted class</i>)	Nilai Kepercayaan (<i>confidence</i>)
1	<i>Apple Cedar/Rust</i>	<i>Apple Cedar/Rust</i>	99.97%
2	<i>Apple Healthy</i>	<i>Apple Healthy</i>	100%
3	<i>Apple Healthy</i>	<i>Apple Scab</i>	98.42%
4	<i>Apple Black Rot</i>	<i>Apple Scab</i>	93.22%
5	<i>Apple Scab</i>	<i>Apple Scab</i>	99.62%
6	<i>Apple Rust</i>	<i>Apple Rust</i>	100%
7	<i>Apple Scab</i>	<i>Apple Scab</i>	100%
8	<i>Apple Healthy</i>	<i>Apple Healthy</i>	100%
9	<i>Apple Scab</i>	<i>Apple Scab</i>	99.99%

Pada tabel diatas kita dapat melihat contoh *dataset* uji terdapat beberapa prediksi yang kurang tepat, untuk lebih memastikan bahwa nilai akurasi pada model diatas benar-benar valid maka penulis melakukan evaluasi model dengan *confusion matrix*, berikut hasilnya penulis gambarkan dalam sebuah *plot confusion matrix*:

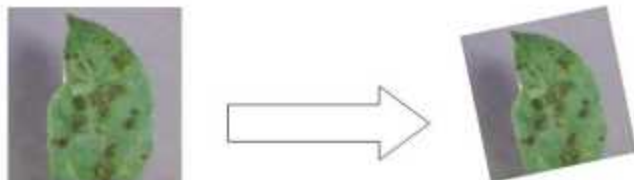


Gambar 4.13. Confusion Matrix CNN dengan data augmentasi *vertical flip*

Dari gambar diatas kita dapat melihat bahwa terdapat beberapa kesalahan prediksi dari 2 kelas yang ada, dimana hasil perhitungan evaluasi *confusion matrix* dengan nilai *test accuracy* 99.39% *precision score* 96.88%, *total recall score* 96.88%, dan *total recall* 96.88%.

c. Pengujian Model CNN Augmentasi Data dengan Teknik *Random Rotation*

Pada skenario ketiga penulis menggunakan teknik augmentasi data *random rotation*. Teknik ini digunakan pada setiap dataset citra dengan cara memutar citra sesuai dengan derajat yang telah ditentukan, pada penelitian ini penulis menentukan besaran derajat sebesar 20° . Berikut contoh dari penerapan teknik *random rotation* pada dataset hama daun apel:



Gambar 4.14. Contoh transformasi rotasi

Dari contoh diatas setiap dataset akan menghasilkan data baru untuk setiap *dataset training*, *data testing* dan *data validation*. Untuk menentukan hasil rotasi suatu objek dapat dengan menggunakan rumus-rumus transformasi geometris seperti dibawah ini:

Tabel 4.5. Tabel Rumus dan Persamaan Transformasi Rotasi

Bentuk Rotasi	Gambar Rotasi	Bentuk Persamaan
Pusat 0 (0,0) sebesar α		$A' = \begin{pmatrix} a' \\ b' \end{pmatrix}$ $= \begin{pmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix}$

Tabel 4.5 (Lanjutan)

Bentuk Rotasi	Gambar Rotasi	Bentuk Persamaan
Pusat (m,n) sebesar α		$A' = \begin{pmatrix} a' \\ b' \end{pmatrix}$ $= \begin{pmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{pmatrix} \begin{pmatrix} a - m \\ b - n \end{pmatrix} + \begin{pmatrix} m \\ n \end{pmatrix}$
pusat $(0,0)$ sebesar α kemudian sebesar β		$A'' = \begin{pmatrix} a'' \\ b'' \end{pmatrix}$ $= \begin{pmatrix} \cos (\alpha + \beta) & -\sin (\alpha + \beta) \\ \sin (\alpha + \beta) & \cos (\alpha + \beta) \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix}$
Rotasi dengan pusat P (m,n) sebesar α kemudian sebesar β		$A'' = \begin{pmatrix} a'' \\ b'' \end{pmatrix}$ $= \begin{pmatrix} \cos (\alpha + \beta) & -\sin (\alpha + \beta) \\ \sin (\alpha + \beta) & \cos (\alpha + \beta) \end{pmatrix} \begin{pmatrix} a - m \\ b - n \end{pmatrix} + \begin{pmatrix} m \\ n \end{pmatrix}$

Dalam pengujian ini berikut potongan *source code* yang digunakan untuk melakukan training data dengan implementasi teknik augmentasi data *Random Rotation* dengan besar jumlah derajat rotasi 20° pada model CNN:

No	Source Code
1	<pre> IMAGE_SIZE = 256 CHANNELS=3 BATCH_SIZE=32 EPOCHS = 60 train_datagen =ImageDataGenerator(#Image Resizing menjadi 256x256 rescale=1./255, rotation_range=20) train_generator = train_datagen.flow_from_directory('dataset2/train', target_size =(IMAGE_SIZE, IMAGE_SIZE), batch_size = 32, class_mode = 'sparse') </pre>
2	<pre> validation_datagen =ImageDataGenerator(rescale=1./255, rotation_range=20) validation_generator= validation_datagen.flow_from_directory('dataset2/val', target_size =(IMAGE_SIZE, IMAGE_SIZE), batch_size = 32, class_mode = 'sparse') </pre>
3	<pre> test_datagen =ImageDataGenerator(rescale=1./255, rotation_range=20) test_generator = test_datagen.flow_from_directory('dataset2/test', target_size =(IMAGE_SIZE, IMAGE_SIZE), batch_size = 32, class_mode = 'sparse') </pre>
4	<pre> input_shape = (IMAGE_SIZE, IMAGE_SIZE, CHANNELS) n_classes =4 </pre>

	<pre> model = models.Sequential([layers.InputLayer(input_shape = input_shape), layers.Conv2D(32, kernel_size=(3, 3), activation='relu'), layers.MaxPooling2D((2, 2)), layers.Conv2D(64, kernel_size=(3, 3), activation='relu'), layers.MaxPooling2D((2, 2)), layers.Conv2D(64, kernel_size=(3, 3), activation='relu'), layers.MaxPooling2D((2, 2)), layers.Conv2D(64, (3, 3), activation='relu'), layers.MaxPooling2D((2, 2)), layers.Conv2D(64, (3, 3), activation='relu'), layers.MaxPooling2D((2, 2)), layers.Flatten(), layers.Dense(64, activation='relu'), layers.Dense(n_classes, activation='softmax'),]) </pre>
5	<pre> model.compile(optimizer='adam', loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=False), metrics=['accuracy']) </pre>

Listing 4.4. Source Code Training Model CNN dengan Augmentasi Data

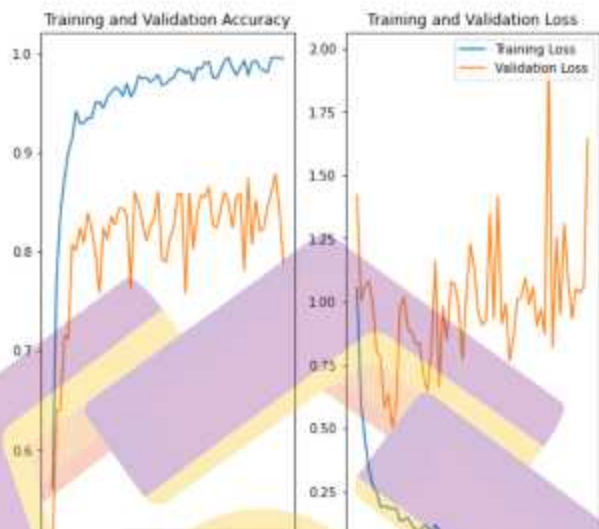
Random Rotation

Keterangan kode program diatas:

- f) Cell 1, 4 Baris pertama adalah konfigurasi parameter dasar meliputi *Image size*, *channels*, *batchsize* dan *epoch*. Pada *cell* ini juga terdapat fungsi untuk membuat *generator data* augmentasi secara teknik rotasi acak (*random rotation*) sebesar 20° dan *re-scale* gambar menjadi 255×255 . Pada bagian *train_generator* adalah sebuah fungsi untuk mengatur sumber dataset yang akan diaugmentasikan.

- g) *Cell 2*, berisi fungsi untuk menggenerate *data validation* dengan augmentasi *random rotation* 20° dan *rescaling* gambar menjadi 256×256 piksel. Kemudian terdapat fungsi untuk menentukan sumber *dataset validation* yang akan di gunakan.
- h) *Cell 3*, fungsi untuk menggenerate *data testing* dengan augmentasi *random rotation* dan *rescaling* gambar menjadi 256×256 piksel. Kemudian terdapat fungsi untuk menentukan sumber *dataset testing* yang akan di gunakan.
- i) *Cell 4*, baris kode ini adalah input model CNN dengan parameter-parameter yang telah ditentukan di awal
- j) *Cell 5*, adalah baris kode untuk mengoptimasi model yang akan di training dengan algoritma *Adam Optimizer*.

Seperti pada pengujian sebelumnya, penulis telah menggunakan parameter yang sama pada model CNN yang akan dibangun. Hasil evaluasi pada model ini menunjukkan *score validation accuracy* sebesar 0.7978 (79.78%) dan *loss validation* sebesar 1.8557(185.57%). Berikut grafik *Training accuracy* dan *Validation Loss* pada skenario ketiga ini:W



Gambar 4.15. Grafik accuracy & loss model CNN dengan teknik augmentasi *Random Rotation*

Pada grafik diatas nilai grafik *training accuracy* terlihat bagus, akan tetapi pada bagian *validation accuracy* cenderung kurang bagus, bahkan terlihat *overfitting*, sedangkan pada bagian *training loss* dengan bertambahnya nilai *epoch* terlihat baik, akan tetapi pada bagian *validation loss* terlihat kurang baik, terjadi *overfitting* cukup tinggi, artinya model dengan teknik ini memiliki tingkat *loss* tinggi. Berikut hasil test prediksi pada *dataset testing* dari model tersebut.

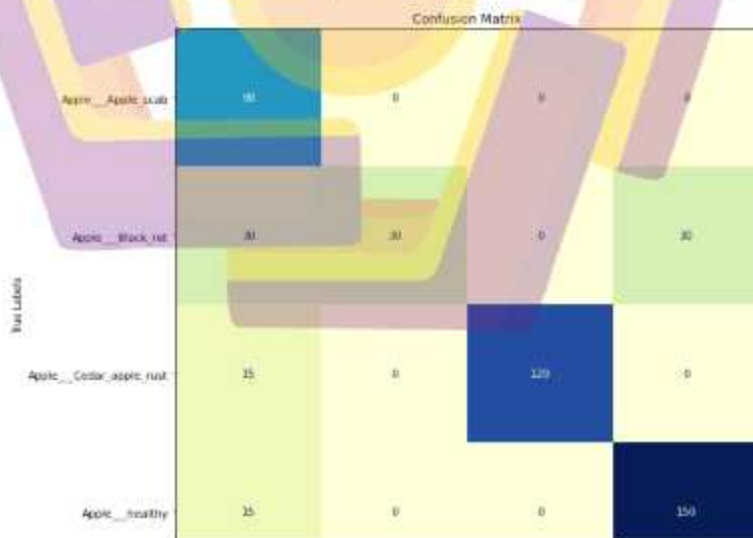
Tabel 4.6. Prediksi Model CNN dengan Teknik Augmentasi Data *Random Rotation*

No	Data Aktual (<i>Actual class</i>)	Prediksi Kelas (<i>predicted class</i>)	Nilai Kepercayaan (<i>confidence</i>)
1	<i>Apple Healthy</i>	<i>Apple Healthy</i>	94,53%

Tabel 4.6. (Lanjutan)

No	Data Aktual (Actual class)	Prediksi Kelas (predicted class)	Nilai Kepercayaan (confidence)
2	<i>Apple Scab</i>	<i>Apple Cedar/Rust</i>	67.41%
3	<i>Apple Scab</i>	<i>Apple Scab</i>	100%
4	<i>Apple Black Rot</i>	<i>Apple Black Rot</i>	99.97%
5	<i>Apple Healthy</i>	<i>Apple Healthy</i>	99.49%
6	<i>Apple Black Rot</i>	<i>Apple Black Rot</i>	99.96%
7	<i>Apple Cedar/Rust</i>	<i>Apple Cedar/Rust</i>	100%
8	<i>Apple Scab</i>	<i>Apple Scab</i>	99.99%
9	<i>Apple Healthy</i>	<i>Apple Healthy</i>	83.5%

Pada tabel prediksi diatas kita dapat melihat contoh prediksi dari kelas-kelas yang ada, dimana terdapat satu prediksi yang bernilai *false negative*, untuk



Gambar 4.16. Confusion Matrix Model CNN Dengan Augmentasi *Random Rotation*

mengetahui uji *dataset testing* secara keseluruhan maka kita dapat melihat hasil *plot confusion matrix* dibawah ini:

Pada gambar plot confusion matrix diatas kita dapat melihat jumlah kebenaran prediksi dari setiap kelas, dimana total nilai dari *train accuracy* 99.51%, *test accuracy* 81.25%, *precisions score* 81.25% dan *total recall score* 81.25%.

d. Pengujian Model CNN Augmentasi Data Dengan Teknkk Kombinasi *Horizontal Flip, Vertical Flip Dan Random Rotation*

Pada skenario ini penulis membangun model dengan dataset yang telah di implementasikan teknik augmentasi, dimana tahap pertama adalah dengan melakukan inisiasi beberapa parameter penting diantaranya adalah image size dengan ukuran 256x256, jumlah *channels* 3, *batch size* 32 dengan jumlah *epoch* 60, kemudian setelah itu penulis menerapkan teknik augmentasi *horizontal flip, vertical flip, rotation range* = 20° , setiap parameter yang ada di terapkan kedalam dataset yang telah dibagi kedalam 80% *data training*, 10% *data testing* dan 10% *data validation*. Semua input konfigurasi tersebut akan diimplementasikan kedalam model CNN yang akan dibangun. Untuk model yang telah dibangun akan dioptimasi dengan algoritma *Adaptive Moment Estimation (Adam Optimizer)*(Kingma & Ba, 2014). ADAM adalah algoritma optimisasi pengganti untuk *stochastic gradient descent* untuk *training model* dalam *deep learning* yang menggabungkan sifat-sifat terbaik dari algoritma *AdaGrad* dan *RMSProp* untuk memberikan algoritma yang lebih optimal dimana dapat menangani gradien yang menyebar dan memiliki *noise*.

Dalam pengujian ini berikut potongan *source code* yang digunakan untuk melakukan training data dengan implementasi teknik augmentasi data gabungan *horizontal flip*, *vertical flip* & *random rotation* dengan besar jumlah drajat rotasi 20^0 pada model CNN:

No	Source Code
1	<pre> IMAGE_SIZE = 256 CHANNELS=3 BATCH_SIZE=32 EPOCHS = 60 train_datagen =ImageDataGenerator(#Image Resizing menjadi 256x256 rescale=1./255, rotation_range=20, Horizontal_flip=True, Vertical_flil=True) train_generator = train_datagen.flow_from_directory('dataset2/train', target_size =(IMAGE_SIZE,IMAGE_SIZE), batch_size = 32, class_mode = 'sparse') </pre>
2	<pre> validation_datagen =ImageDataGenerator(rescale=1./255, rotation_range=20, Horizontal_flip=True, Vertical_flil=True) validation_generator= validation_datagen.flow_from_directory('dataset2/val', target_size =(IMAGE_SIZE,IMAGE_SIZE), batch_size = 32, class_mode = 'sparse') </pre>
3	<pre> test_datagen =ImageDataGenerator(rescale=1./255, rotation_range=20, Horizontal_flip=True, Vertical_flil=True) </pre>

	<pre> test_generator = test_datagen.flow_from_directory('dataset2/test', target_size =(IMAGE_SIZE,IMAGE_SIZE), batch_size = 32, class_mode = 'sparse') </pre>
4	<pre> input_shape = (IMAGE_SIZE,IMAGE_SIZE,CHANNELS) n_classes =4 model = models.Sequential([layers.InputLayer(input_shape = input_shape), layers.Conv2D(32, kernel_size=(3,3), activation='relu'), layers.MaxPooling2D((2,2)), layers.Conv2D(64, kernel_size=(3,3), activation='relu'), layers.MaxPooling2D((2,2)), layers.Conv2D(64, kernel_size=(3,3), activation='relu'), layers.MaxPooling2D((2,2)), layers.Conv2D(64, (3, 3), activation='relu'), layers.MaxPooling2D((2,2)), layers.Conv2D(64, (3, 3), activation='relu'), layers.MaxPooling2D((2, 2)), layers.Flatten(), layers.Dense(64, activation='relu'), layers.Dense(n_classes, activation='softmax'),]) </pre>
5	<pre> model.compile(optimizer='adam', loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=False), metrics=['accuracy']) </pre>

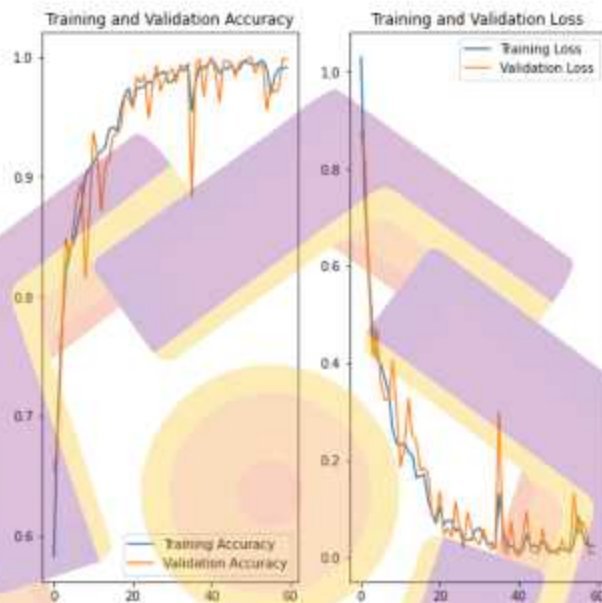
Listing 4.5. Source Code Training Model CNN dengan Augmentasi Data

Gabungan

Keterangan kode program diatas:

- k) *Cell 1*, 4 Baris pertama adalah konfigurasi parameter dasar meliputi *Image size*, *channels*, *batchsize* dan *epoch*. Pada *cell* ini juga terdapat fungsi untuk membuat *generator data* augmentasi dengan teknik gabungan yaitu teknik *horizontal flip*, *vertical flip* dan *random rotation* dan *me-rescale* gambar menjadi 255x255. Pada bagian *train_generator* adalah sebuah fungsi untuk mengatur sumber dataset yang akan diaugmentasikan.
- l) *Cell 2*, berisi fungsi untuk menggenerate *data validation* dengan augmentasi *random rotation 20°* dan *rescaling* gambar menjadi 256x256 piksel. Kemudian terdapat fungsi untuk menentukan sumber *dataset validation* yang akan di gunakan.
- m) *Cell 3*, fungsi untuk menggenerate *data testing* dengan augmentasi gabungan dan *rescaling* gambar menjadi 256x256 piksel. Kemudian terdapat fungsi untuk menentukan sumber *dataset testing* yang akan di gunakan.
- n) *Cell 4*, baris kode ini adalah input model CNN dengan parameter-parameter yang telah ditentukan di diawal
- o) *Cell 5*, adalah baris kode untuk mengoptimasi model yang akan di training dengan algoritma *Adam Optimizer*.

Hasil evaluasi model pada skenario pertama ini memiliki *validation accuracy* akurasi sebesar 0.9957 (99.57%) dengan nilai *loss validation* 0.0355 (0.3%).



Gambar 4.17. Grafik *accuracy & loss model CNN Dengan Teknik Augmentasi Gabungan*

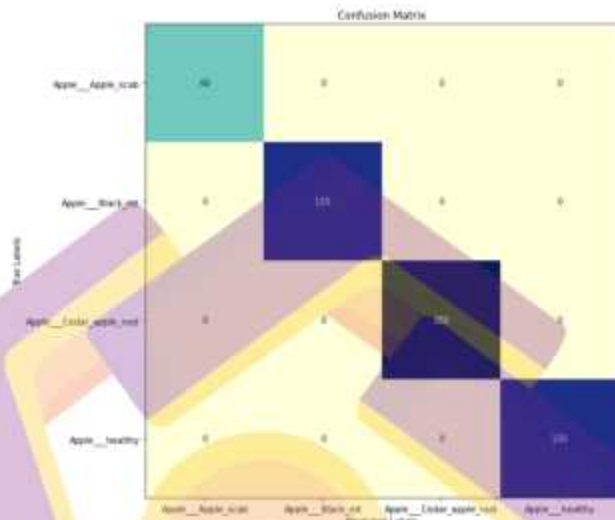
Berikut Grafik *training accuracy* dan *validation accuracy*:

Grafik diatas menunjukkan bahwa pergerakan training dengan beberapa penyesuaian parameter pada model yang dibangun cukup baik, jika kita lihat dari grafik pada beberapa epoch tingkat akurasi bisa sampai 100% dan *overfitting* pada grafik ini dapat direduksi dengan baik. Berikut hasil test prediksi pada *dataset testing* dari model tersebut.

Tabel 4.7 Tabel Prediksi Model CNN dengan teknik augmentasi data gabungan

No	Data Aktual (<i>Actual class</i>)	Prediksi Kelas (<i>predicted class</i>)	Nilai Kepercayaan (<i>confidence</i>)
1	<i>Apple Black Root</i>	<i>Apple Black Root</i>	99.79%
2	<i>Apple Healthy</i>	<i>Apple Healthy</i>	100%
3	<i>Apple Scab</i>	<i>Apple Scab</i>	99.98%
4	<i>Apple Healthy</i>	<i>Apple Healthy</i>	100%
5	<i>Apple Black Rot</i>	<i>Apple Black Rot</i>	100%
6	<i>Apple Cedar/Rust</i>	<i>Apple Cedar/Rust</i>	100%
7	<i>Apple Healthy</i>	<i>Apple Healthy</i>	99.98%
8	<i>Apple Black Rot</i>	<i>Apple Black Rot</i>	100%
9	<i>Apple Cedar/Rusti</i>	<i>Apple Cedar/Rust</i>	100%

Dari contoh implementasi model pada grafik dan *dataset testing* diatas kita dapat melihat tingkat akurasi dan *confidence* dari model CNN yang dibangun cukup akurat. Untuk memastikan bahwa nilai akurasi pada model diatas benar-benar valid maka penulis melakukan evaluasi model dengan *confusion matrix*, berikut hasilnya penulis gambarkan dalam sebuah plot *confusion matrix*:



Gambar 4.18. *Plot Confusion Matrix Model CNN dengan augmentasi gabungan*

Dari gambar diatas kita dapat melihat bahwa seluruh prediksi *data testing* pada model ini seluruhnya benar dengan nilai perhitungan *train accuracy 99.13%*, *test accuracy precision score 100%*, *total recall score 100%* dan *total-recall 100%*.

4.3. Evaluasi Model

Pada penelitian ini penulis melakukan beberapa pengujian dengan beberapa parameter yang diujikan pada model yang dibangun, hasil evaluasi dari setiap model yang dibangun terdapat parameter-parameter yang berpengaruh terhadap tingkat akurasi model diantaranya adalah sebagai berikut

4.3.1. Pengaruh Ukuran Citra

Penulis melakukan percobaan pengujian dengan data augmentasi gabungan dan input citra dengan dua ukuran yang berbeda, ukuran citra yang diujikan pada model ini adalah dengan ukuran 256x256 piksel dan 512x512 piksel dengan jumlah *epoch* pada percobaan ini adalah 60. Berikut hasil uji validasi akurasi citra tersebut:

Tabel 4.8. Tabel Perbandingan *accuracy & loss* berdasarkan ukuran citra

Ukuran Citra	Validasi Akurasi	Validasi Loss	Waktu Pelatihan
256x256	99,57%	3,55%	1.9 Jam
512x512	99,14%	1,70%	21,7 Jam

Berdasarkan tabel perbandingan diatas kita dapat melihat perbedaan dimana ukuran citra berpengaruh kepada waktu *training* dimana semakin besar ukuran citra berpengaruh kepada waktu, validasi akurasi dan validasi *loss* .

4.3.2. Pengaruh Jumlah *Epoch*

Salah satu parameter yang mempengaruhi performa sebuah model CNN adalah jumlah *epoch*, pengertian dari *epoch* itu sendiri adalah satu siklus putaran dari algoritma *machine learning* belajar dari *dataset training*. proses pembelajaran berulang-ulang bertujuan untuk mencapai nilai konvergensi yang baik, dimana tidak ada patokan yang pasti mengenai jumlah *epoch* yang harus di gunakan(Wibawa, 2017), maka dari itu penulis melakukan uji coba parameter *epoch* dari rentang 10 sampai dengan 60. Berikut hasil uji model CNN dengan

beberapa perubahan parameter *epoch*. Berikut hasil uji parameter jumlah *epoch*:

Tabel 4.9. Perbandingan Akurasi dan loss berdasarkan jumlah epoch

<i>Epoch</i>	Validasi Akurasi	Validasi Loss	Waktu
10	76.07%	25.02%	10 Menit
20	92.90%	28.06%	40 Menit
30	94.83%	16.15%	57.5 Menit
40	98.92%	4.33%	76.6 Menit
50	98.70%	4.22%	104.1 Menit
60	99.57%	2.20%	115 Menit
75	99.57%	1.27%	162.5 Menit
80	99.57%	2.38%	173.4 Menit

Berdasarkan data diatas kita dapat melihat pengaruh jumlah *epoch* pada model yang dibangun cukup berpengaruh pada waktu training dan akurasinya, sehingga perlu adanya percobaan-percobaan mengenai penentuan jumlah epoch yang sesuai dengan model yang kita bangun. Pada penelitian ini jumlah epoch yang memiliki tingkat akurasi tinggi dan loss yang rendah adalah ada pada *epoch* 75 dengan nilai validasi akurasi 99.57% stagnan dari epoch ke-60 dan ke-80. Sehingga dapat diambil kesimpulan bahwa jumlah epoch mempengaruhi akurasi pada penelitian ini dengan nilai validasi tertinggi stagnan 99.57%, jumlah epoch juga berpengaruh pada durasi training modelnya dan cukup membebani komputer.

4.3.3. Perbandingan dengan penelitian sebelumnya

Sebagai bahan perbandingan dengan penelitian sebelumnya berikut penulis sajikan beberapa penelitian serupa dengan teknik-teknik augmentasi serta algoritma yang digunakan dengan objek penelitian yang berbeda:

Tabel 4.10. Komparasi dengan penelitian terdahulu

No	Author	Objek Penelitian	Data Augmentasi	Algoritma	Akurasi
1.	(Enkvetchakul & Surinta, 2021)	Klasifikasi Hama Pada Daun Singkong	Kombinasi Random Rotation, Shift	CNN dengan arsitektur MobilenetV2 dan NASnetMobile	96.15%
2.	(Baranwal dkk., 2019)	Klasifikasi Hama Pada Daun Apel	Horizontal Flip, Vertical Flip, Width Shift, Height Shift, Rotation	CNN Arsitektur GoogleNet	98.42%
3.	(Aluru dkk., t.L)	Klasifikasi hama pada daun tomat	Rotation, horizontal flipping, vertical & horizontal shift	CNN Arsitektur LeNet	94.85%
4.	(Safdar dkk., 2020)	Komparasi analisis Data Citra Magnetic Resonance Image (MRI)	Rotation 180° dan Rotation 90°	You Only Look Once (YOLO) V3	96%
5	(Mega Pradnya & Putri Kusumaningtyas, t.L)	Klasifikasi bumbu dapur	Rescale, Shift, Share_range, Rotation, fill Mode	CNN	80%
6	Penelitian Terbaru (penulis)	Klasifikasi Hama Pada Daun Apel	Kombinasi Random Rotation, Horizontal Flip, vertical Flip	CNN	99,57 %

BAB V

PENUTUP

5.1. Kesimpulan

Berdasarkan hasil penelitian diatas maka dapat penulis menyimpulkan bahwa:

- a. Tingkat akurasi pada model *Convolutional Neural Network* sebelum di terapkan teknik augmentasi yang tepat memiliki tingkat akurasi lebih rendah dimana hasil evaluasi dengan *confusion matrix* model ini memiliki training *total test accuracy* 90.62%, *total precision score* 90.62% dan *total recall score* 90.62%. Sedangkan pada model *Convolutional Neural Network* yang telah di implementasikan teknik augmentasi data yang tepat (teknik augmentasi gabungan) hasil evaluasi dengan *confusion matrix* memiliki *train accuracy* 99.13%, *test accuracy precision score* 100%, *total recall score* 100% dan *total recall* 100%.
- b. Teknik augmentasi data yang diimplementasikan pada model *Convolutional Neural Network* adalah teknik augmentasi data transformasi geometris diantaranya adalah teknik *horizontal flip* dengan nilai *test accuracy precision score* 90.62%, *total recall score* 90.62% dan *total recall* 90.62%. kemudian *vetical flip* dengan nilai *test accuracy* 99.39% *precision score* 96.88%, *total recall score* 96.88%, dan *total recall* 96.88%, kemudian *random rotation* dengan nilai *train accuracy* 99.51%, *test accuracy* 81.25%, *precisions score* 81.25% dan *total recall score* 81.25%, dan yang terakhir adalah teknik augmentasi gabungan memiliki nilai akurasi dan evaluasi *confusion matrix*

terbaik yaitu *train accuracy* 99.13%, *test accuracy precision score* 100%, *total recall score* 100% dan *total recall* 100%

- c. Terdapat beberapa hal yang mempengaruhi tingkat akurasi dan waktu training diantaranya adalah jumlah *epoch* dan ukuran citra, dimana semakin banyak jumlah *epoch* ataupun semakin besar gambar maka akan berpengaruh kepada waktu training data.

5.2. Saran

Berdasarkan penelitian yang telah penulis lakukan berikut beberapa saran untuk pengembangan penelitian ini, diantaranya adalah sebagai berikut:

- a. Pada penelitian ini dataset yang digunakan untuk studi kasus baru satu jenis yaitu dataset citra hama daun apel. Kedepan diharapkan dapat menambahkan jenis dataset hama pada tanaman yang lebih beragam.
- b. Pada penelitian ini data hama yang didapatkan baru 3 jenis hama yang dapat di deteksi, harapanya kedepan dapat ditambahkan dengan dataset hama pada daun apel dengan jenis hama yang baru yang telah terdeteksi oleh para peneliti dibidang yang terkait
- c. Teknik Augmentasi data yang dilakukan pada model CNN yang dibangun bisa ditambahkan dengan teknik-teknik yang lainnya.
- d. Output penelitian ini terbatas sampai pemodelan *machine learning*, harapanya kedepan dapat dikembangkan dengan mengimplementasikan model yang dibangun kedalam *platform* yang ada agar dapat langsung di manfaatkan oleh masyarakat.

DAFTAR PUSTAKA

PUSTAKA BUKU

Suyanto. (2018). *Machine Learning Tingkat Dasar dan Lanjut* (Vol. 1). Informatika.

PUSTAKA MAJALAH, JURNAL ILMIAH ATAU PROSIDING

Abayomi-Alli, O. O., Damaševičius, R., Misra, S., & Maskeliūnas, R. (2021). Cassava disease recognition from low-quality images using enhanced data augmentation model and deep learning. *Expert Systems*, 38(7). <https://doi.org/10.1111/exsy.12746>

Aluru, S., Jaypee Institute of Information Technology University, University of Florida. College of Engineering, IEEE Computer Society, IEEE Computer Society, Technical Committee on Parallel Processing, & Institute of Electrical and Electronics Engineers. (t.t.). *2018 Eleventh International Conference on Contemporary Computing (IC3): 2-4 August 2018, Jaypee Institute of Information Technology, Noida, India.*

Amity University, & Institute of Electrical and Electronics Engineers. (2020). *10th International Conference on Cloud Computing, Data Science & Engineering : proceedings of the Confluence 2020 : 29-31 January 2020, Amity University, Uttar Pradesh, India.*

Amity University, Institute of Electrical and Electronics Engineers, & Institute of Electrical and Electronics Engineers. United Kingdom and Republic of Ireland Section. (2020). *2020 International Conference on Intelligent Engineering and Management (ICIEM-2020).*

- Asrafil, A., Paliwang, A., Ridwan, M., Septian, D., Cahyanti, M., Ericks, D., Swedia, R., & Informatika, J. T. (2021). *klasifikasi penyakit tanaman apel dari citra daun dengan convolutional neural network*.
- Baranwal, S., Khandelwal, S., & Arora, A. (2019). *Deep Learning Convolutional Neural Network for Apple Leaves Disease Detection under responsibility of International Conference on Sustainable Computing in Science, Technology and Management*.
<https://ssrn.com/abstract=3351641>
- Bowles, C., Chen, L., Guerrero, R., Bentley, P., Gunn, R., Hammers, A., Dickie, D. A., Hernández, M. V., Wardlaw, J., & Rueckert, D. (2018). *GAN Augmentation: Augmenting Training Data using Generative Adversarial Networks*. <http://arxiv.org/abs/1810.10863>
- Dadang. (2010, Desember 20). *Konsep Hama dan Dinamika Populasi. Potensi Kerusakan Dan Teknik Pengendaliannya*.
- DeVries, T., & Taylor, G. W. (2017). *Dataset Augmentation in Feature Space*.
<http://arxiv.org/abs/1702.05538>
- Dhamayanti, R., Rohmah, M. F., & Zahara, S. (2021). *Penggunaan Deep Learning Dengan Metode Convolutional Neural Network Untuk Klasifikasi Kualitas Kol Berdasarkan Citra Fisik. SUBMIT (Jurnal Ilmiah Teknologi Informasi Dan Sains)*, 1, 8-15.
<http://ejurnal.unim.ac.id/index.php/submit>
- Enkvetchakul, P., & Surinta, O. (2021). *Effective Data Augmentation and Training Techniques for Improving Deep Learning in Plant Leaf Disease*

- Recognition. *Applied Science and Engineering Progress*.
<https://doi.org/10.14416/j.asep.2021.01.003>
- Hasan Mahmud, K., & al Faraby, S. (2019). *Klasifikasi Citra Multi-Kelas Menggunakan Convolutional Neural Network*.
- Inoue, H. (2018). *Data Augmentation by Pairing Samples for Images Classification*. <http://arxiv.org/abs/1801.02929>
- Jing, Y., Yang, Y., Feng, Z., Ye, J., Yu, Y., Song, M., & Member, S. (2019). *Neural Style Transfer: A Review*. <https://osf.io/f8tu4/>.
- Kingma, D. P., & Ba, J. (2014). *Adam: A Method for Stochastic Optimization*. <http://arxiv.org/abs/1412.6980>
- Liu, B., Zhang, Y., He, D. J., & Li, Y. (2018). Identification of apple leaf diseases based on deep convolutional neural networks. *Symmetry*, 10(1).
<https://doi.org/10.3390/sym10010011>
- Mega Pradnya, W. D., & Putri Kusumaningtyas, A. (t.t.). *JURNAL MEDIA INFORMATIKA BUDIDARMA Analisis Pengaruh Data Augmentasi Pada Klasifikasi Bumbu Dapur Menggunakan Convolutional Neural Network*. 6, 2022–2031. <https://doi.org/10.30865/mib.v6i4.4201>
- Michelle Grabowski. (2019). *Balck Rot Apple*. Michelle Grabowski
- Mishra, S., Sachan, R., & Rajpal, D. (2020). Deep Convolutional Neural Network based Detection System for Real-time Corn Plant Disease Recognition. *Procedia Computer Science*, 167, 2003–2010.
<https://doi.org/10.1016/j.procs.2020.03.236>

Olga Chernytska. (t.t.). *Complete Guide to Data Augmentation for Computer Vision*. Diambil 21 November 2022, dari Complete Guide to Data Augmentation for Computer Vision

Rebecca Koetter, & Michelle Grabowski. (2019). *Apple Scab of Apples and Crabapples*. <https://extension.umn.edu/plant-diseases/apple-scab#cultural-practices-1165362>

Rozaqi, A. J., Sunyoto, A., & Arief, R. (2021). Deteksi Penyakit pada Daun Kentang Menggunakan Pengolahan Citra dengan Metode Convolutional Neural Network Detection of Potato Leaves Disease Using Image Processing with Convolutional Neural Network Methods. *Citec Journal*, 8(1), 22–31.

Safdar, M. F., Alkobaisi, S. S., & Zahra, F. T. (2020). A comparative analysis of data augmentation approaches for magnetic resonance imaging (MRI) scan images of brain tumor. *Acta Informatica Medica*, 28(1), 29–36. <https://doi.org/10.5455/AIM.2020.28.29-36>

Summers, C., & Dinneen, M. J. (2019). Improved mixed-example data augmentation. *Proceedings - 2019 IEEE Winter Conference on Applications of Computer Vision, WACV 2019*, 1262–1270. <https://doi.org/10.1109/WACV.2019.00139>

Suyanto. (2018). *Machine Learning Tingkat Dasar dan Lanjut* (Vol. 1). Informatika.

Wibawa, M. S. (2017). Pengaruh Fungsi Aktivasi, Optimisasi dan Jumlah Epoch Terhadap Performa Jaringan Saraf Tiruan. *Jurnal Sistem Dan*

Wicaksono, G., & Andryana, S. (2018). Aplikasi Pendeteksi Penyakit Pada Daun Tanaman Apel Dengan Metode Convolutional Neural Network. *JOINTECS (Journal of Information Technology and Computer Science)*, 5(1), 9–16.

Zhang, Q. shi, & Zhu, S. chun. (2018). Visual interpretability for deep learning: a survey. Dalam *Frontiers of Information Technology and Electronic Engineering* (Vol. 19, Issue 1, hlm. 27–39). Zhejiang University. <https://doi.org/10.1631/FITEE.1700808>

Zhong, Z., Zheng, L., Kang, G., Li, S., & Yang, Y. (2017). *Random Erasing Data Augmentation*. <https://github.com/zhunzhong07/Random-Erasing>.

PUSTAKA ELEKTRONIK

Badan Pusat Statistik Nasional. (2020). *Produksi Tanaman Buah-buahan Tahun 2020*. <https://www.bps.go.id/indicator/55/62/2/produksi-tanaman-buah-buahan.html>

Badan Pusat Statistik Nasional. (2021). *Produksi Tanaman Buah-buahan 2021*. <https://www.bps.go.id/indicator/55/62/1/produksi-tanaman-buah-buahan.html>

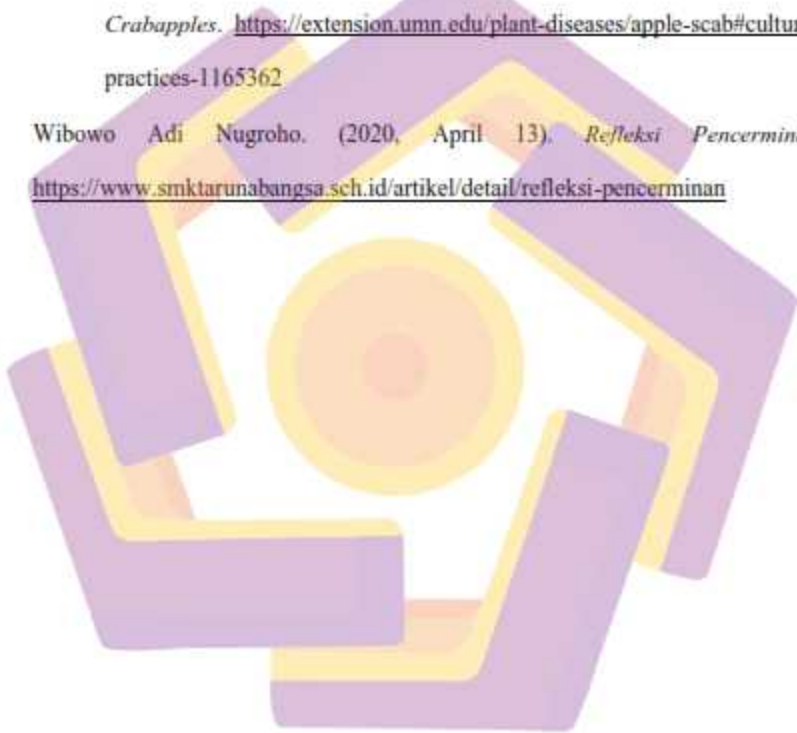
Wibowo Adi Nugroho. (2020, April 13). *Refleksi Pencerminan*. <https://www.smktarunabangsa.sch.id/artikel/detail/refleksi-pencerminan>

Jennifer Olson. (2017, Februari). *Cedar-Apple Rust*.

<https://extension.okstate.edu/fact-sheets/cedar-apple-rust.html#:~:text=Cedar%2Dapple%20rust%20and%20similar,result%20may%20be%20tree%20death>.

Rebecca Koetter, & Michelle Grabowski. (2019). *Apple Scab of Apples and Crabapples*. <https://extension.umn.edu/plant-diseases/apple-scab#cultural-practices-1165362>

Wibowo Adi Nugroho. (2020, April 13). *Refleksi Pencerminan*. <https://www.smktarunabangsa.sch.id/artikel/detail/refleksi-pencerminan>



LAMPIRAN

