

KLASIFIKASI PENYAKIT DAUN TEH MENGGUNAKAN DENSENET-121

LAPORAN TUGAS AKHIR

Oleh:

Haikal Efendi

105219027



**FAKULTAS SAINS DAN ILMU KOMPUTER
PROGRAM STUDI ILMU KOMPUTER
UNIVERSITAS PERTAMINA**

2023



KLASIFIKASI PENYAKIT DAUN TEH MENGGUNAKAN DENSENET-121

LAPORAN TUGAS AKHIR

Oleh:

Haikal Efendi

105219027



**FAKULTAS SAINS DAN ILMU KOMPUTER
PROGRAM STUDI ILMU KOMPUTER
UNIVERSITAS PERTAMINA
AGUSTUS 2023**

- Dikaitkan Cipta milik Universitas Pertamina*
- Dilindungi Undang-Undang*
- © Copyright of Universitas Pertamina*
1. Dilarang mengutip karya tulis ini, kecuali:
 - a. menyebutkan sumber sesuai kaidah kecendekiaan;
 - b. pengutipan hanya untuk keperluan pendidikan, penulisan karya ilmiah atau penelitian;
 - c. pengutipan tidak merugikan Universitas Pertamina.
 2. Dilarang mempublikasikan atau memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa adanya izin dari Universitas Pertamina.





Universitas
Pertamina

2. a. menyebutkan sumber sesuai kaidah kecendekiaan;
b. pengutipan hanya untuk keperluan pendidikan, penulisan karya ilmiah atau penelitian;
c. pengutipan tidak merugikan Universitas Pertamina.

Dilarang mempublikasikan atau memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa adanya izin dari Universitas Pertamina.

LEMBAR PENGESAHAN

Judul Tugas Akhir

KLASIFIKASI PENYAKIT DAUN TEH MENG GUNAKAN DENSENET-121

Nama Mahasiswa

: Haikal Efendi

Nomor Induk Mahasiswa

: 105219027

Program

: Ilmu Komputer

Eakultas

Jakarta, 30 Agustus 2023

MENGESAHKAN

Pembimbing I

Pembimbing II

Dr. Eng. Muhammad Abdillah
NIP.116153

Dr. Eng. Muhammad Abdillah
NIP/116153

MENGETAHUI,

Ketua Program Studi



Ade Irawan, Ph.D.

Ade Irawati, Ph.D.

NIP. 116130

LEMBAR PERNYATAAN

Dengan ini saya menyatakan bahwa Tugas Akhir berjudul **KLASIFIKASI PENYAKIT DAUN TEH MENGGUNAKAN DENSENET-121** ini adalah benar-benar merupakan hasil karya saya sendiri dan tidak mengandung materi yang ditulis oleh orang lain kecuali telah dikutip sebagai referensi yang sumbernya telah dituliskan secara jelas sesuai dengan kaidah penulisan karya ilmiah.

Apabila dikemudian hari ditemukan adanya kecurangan dalam karya ini, saya bersedia menerima sanksi dari Universitas Pertamina sesuai dengan peraturan yang berlaku.

Demi pengembangan ilmu pengetahuan, saya menyetujui untuk memberikan kepada Universitas Pertamina hak bebas royalti noneksklusif (*non-exclusive royalty-free right*) atas Tugas Akhir ini beserta perangkat yang ada. Dengan hak bebas royalti noneksklusif ini Universitas Pertamina berhak menyimpan, mengalih media/format-kan, mengelola dalam bentuk pangkatan data (*database*), merawat, dan mempublikasikan Tugas Akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya

30
Jakarta, Agustus 2023

Yang membuat pernyataan,



Haikal Efendi



ABSTRAK

Haikal Efendi. 105219027. KLASIFIKASI PENYAKIT DAUN TEH MENGGUNAKAN DENSENET-121.

Teh adalah salah satu tanaman yang cukup penting bagi kesehatan dan sangat sering dikonsumsi oleh banyak orang. Perawatan dan pencegahan pada tanaman teh dibutuhkan agar dapat diproduksi kemudian dikonsumsi. Penelitian ini ditujukan untuk mendapatkan model klasifikasi penyakit daun teh dan mendapatkan akurasi model yang baik. Penelitian ini dilakukan dengan cara mengumpulkan data terlebih dahulu yang didapatkan dari Mendeley Data. Dataset yang didapatkan berupa data gambar daun teh yang diklasifikasi ke dalam 3 penyakit yaitu Tea Red Leaf Spot, Tea Leaf Blight, dan Tea Red Scab. Model Arsitektur DenseNet digunakan pada penelitian ini dengan harapan dapat memproses gambar untuk mendapatkan hasil akurasi yang baik agar dapat mendeteksi dan mengklasifikasi penyakit pada gambar daun teh dengan baik. Hasil dari klasifikasi menggunakan DenseNet-121 ini cukup baik yaitu akurasi 93.33%. Hasil klasifikasi ini dapat memprediksi dengan benar 56 gambar dari 60 data *test* model.

Kata kunci: Daun teh, CNN, DenseNet, klasifikasi

Universitas Pertamina
dilindungi undang-undang
penyalahgunaan
dilakukan
sesuai
kecendekiaan;

2. Dilarang mempublikasikan atau memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun
- c. pengutipan tidak merugikan Universitas Pertamina.

ABSTRACT

Haikal Efendi. 105219027. TEA PLANT DISEASE CLASSIFICATION USING DENSENET-121.

Tea is a plant that is quite important for health and is often consumed by many people. Care and prevention of tea plants is needed so that they can be produced and then consumed. This research is aimed at obtaining a classification model for tea leaf diseases and obtaining good model accuracy. This research was carried out by first collecting data obtained from Mendeley Data. The dataset obtained is in the form of image data of tea leaves which are classified into 3 diseases, namely Tea Red Leaf Spot, Tea Leaf Blight, and Tea Red Scab. The DenseNet Architectural Model is used in this study with the hope that it can process images to obtain good accuracy results in order to detect and classify diseases in tea leaf images properly. The results of the classification using DenseNet-121 are quite good, namely an accuracy of 93.33%. The results of this classification can correctly predict 56 images from 60 model test data.

Keywords: Tea leaf, CNN, network, DenseNet, classification



Hak Cipta milik Universitas Pertamina. Dilarang menyebarkan karya tulis ini, kecuali:

© Copyright of Universitas Pertamina

- c. pengutipan tidak merugikan Universitas Pertamina;
2. Dilarang mempublikasikan atau memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa adanya izin dari Universitas Pertamina.



KATA PENGANTAR

Penulis mengucapkan puji syukur kepada Allah SWT atas bimbingan-Nya dalam menyelesaikan skripsi ini sebagai syarat untuk memperoleh gelar Sarjana Ilmu Komputer di Universitas Pertamina. Penulis mengucapkan terima kasih kepada semua yang telah membantu dan mendukung, termasuk dosen pembimbing (Ibu Dr. Tasmi, S.Si, M.Si dan Pak Dr. Eng. Muhammad Abdillah, S.T., M.T.), keluarga, teman-teman seangkatan dan sejurusan (khususnya Zhafran Ramadhan dan Hernawan Alfian), serta sumber referensi yang menjadi acuan dalam penulisan. Penulis menyadari adanya kekurangan dalam skripsi ini dan siap menerima kritik, saran, dan masukan yang konstruktif untuk perbaikan di masa depan.

Akhir kata, penulis berharap bahwa skripsi ini dapat memberikan manfaat dan kontribusi yang positif dalam bidang Klasifikasi Penyakit Daun Teh Menggunakan Convolutional Neural Network. Semoga skripsi ini dapat menjadi sumber inspirasi dan referensi bagi peneliti dan pembaca lainnya.

Terima kasih.

Jakarta, 10 Agustus 2023

Haikal Efendi

2. Dilarang mempublikasikan atau memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa adanya izin dari Universitas Pertamina.
- c. pengutipan tidak mengikan Universitas Pertamina.

DAFTAR ISI

KATA PENGANTAR	iii
DAFTAR ISI	iv
DAFTAR TABEL	vi
DAFTAR GAMBAR	vii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	2
1.4 Tujuan Penelitian	2
1.5 Manfaat Penelitian	2
BAB II TINJAUAN PUSTAKA	4
2.1 Penyakit Daun Teh	4
2.2 Pembelajaran Mesin (Machine Learning)	4
2.3 Convolutional Neural Network (CNN)	5
2.4 Membangun Blok Arsitektur CNN	6
2.5 DenseNet	6
2.6 Laporan Klasifikasi	8
2.7 Penelitian Terkait	8
BAB III METODE PENELITIAN	11
3.1 Alur Penelitian	11
3.2 Pengumpulan Data	11
3.3 Pra-pemrosesan Data	11
3.4 Pembangunan Model DenseNet	12
3.5 Pelatihan Model	12
3.6 Evaluasi Model	12

3.7 Pengujian Lanjutan	12
BAB IV HASIL DAN PEMBAHASAN	15
4.1 Proses Pengolahan Data	15
4.2 Hasil dan Pembahasan	18
4.2.1 Percobaan Epoch = 50	18
4.2.2 Percobaan Epoch = 100	22
4.2.3 Percobaan Epoch = 300	26
4.2.4 Ringkasan	28
BAB V KESIMPULAN DAN SARAN	30
5.1 Kesimpulan	30
5.2 Saran	30
DAFTAR PUSTAKA	32

Universitas Pertamina

 Universitas Pertamina
 Dilarang menggkop karya tulis ini, kecuali:
 1. menyertakan surat sesuai kaidah kecendekiaan;
 b. pengutipan hanya untuk keperluan pendidikan, penulisan karya ilmiah atau penelitian;
 c. pengutipan tidak merugikan Universitas Pertamina.
 2. Dilarang mempublikasikan atau memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa adanya izin dari Universitas Pertamina.



DAFTAR TABEL

Tabel 4.1	Evaluasi <i>Score</i>	18
Tabel 4.2	Laporan Evaluasi	20
Tabel 4.3	Hasil Prediksi	21
Tabel 4.4	Evaluasi <i>Score</i>	22
b. Tabel 4.5	Laporan Evaluasi <i>Epoch</i> =100	23
Tabel 4.6	Hasil Prediksi	23
Tabel 4.7	Evaluasi <i>Score</i>	26
Tabel 4.8	Laporan Evaluasi <i>Epoch</i> =300	26
Tabel 4.9	Hasil Prediksi <i>Epoch</i> =300	27
Tabel 4.10	Ringkasan per <i>Epoch</i>	28

Universitas Pertamina
Dilindungi Undang-Undang
Penyalahgunaan dan pengutipan tulis ini, kecuali:
1. Ter sesuai kaidah kecendekiaan;
2. Dilarang mempublikasikan atau memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun
tanpa adanya izin dari Universitas Pertamina.

DAFTAR GAMBAR

Gambar 2.1	Array of numbers (Yamashita et al., 2018)	6
Gambar 2.2	Alur Arsitektur DenseNet (Huang et al., 2017)	7
Gambar 3.1	Alur Proses Klasifikasi Gambar	11
Gambar 4.1	Ilustrasi Pembagian Data	15
Gambar 4.2	Evaluasi Score (25)	18
Gambar 4.3	Evaluasi Score (50)	18
Gambar 4.4	Prediksi Label Pertama	19
Gambar 4.5	Prediksi 9 Gambar	19
Gambar 4.6	Gambar yang Salah Diprediksi	21
Gambar 4.7	Gambar yang Salah Diprediksi (2)	21
Gambar 4.8	Evaluasi Score (50)	22
Gambar 4.9	Evaluasi Score (100)	22
Gambar 4.10	Gambar yang Salah Diprediksi	24
Gambar 4.11	Gambar yang Salah Diprediksi (2)	24
Gambar 4.12	Gambar yang Salah Diprediksi (3)	25
Gambar 4.13	Gambar yang Salah Diprediksi (4)	25
Gambar 4.14	Evaluasi Score (150)	26
Gambar 4.15	Evaluasi Score (300)	26
Gambar 4.16	Gambar yang Salah Diprediksi	27
Gambar 4.17	Gambar yang Salah Diprediksi (2)	28



BAB I

PENDAHULUAN

1.1 Latar Belakang

Teh merupakan tanaman yang penting bagi kesehatan dan dikonsumsi oleh banyak orang. Adapun manfaat dari teh bagi kesehatan seperti melawan radikal bebas, membantu menurunkan berat badan, membuat kulit lebih sehat, menurunkan risiko penyakit *Alzheimer*, menurunkan risiko diabetes, dan lain - lain. Di dalam teh juga terdapat zat yang bagus dan bermanfaat seperti vitamin E, vitamin C, vitamin A, dan Polifenol (Kesehatan, 2022).

Tanaman teh sangat dibutuhkan oleh masyarakat, sehingga perkebunan teh perlu dijaga dan dilestarikan. Tantangan terbesar bagi petani adalah adanya penyakit daun teh yang dapat merusak kualitas daun. Selain mempengaruhi kualitas, teh yang terkena penyakit dapat menyebabkan kerugian ekonomi yang cukup serius bagi petani (Bao et al., 2022). Terdapat banyak jenis penyakit daun teh diantaranya *Tea Leaf Blight*, *Tea Red Leaf Spot*, dan *Tea Red Scab*. *Tea Leaf Blight* adalah penyakit yang menular pada daun teh yang ditandai dengan perubahan warna daun berubah menjadi coklat, untuk *Tea Red Leaf Spot* yaitu penyakit bercak pada daun yang biasanya berwarna coklat kemerah-merahan dan *Tea Red Scab* adalah penyakit yang memiliki bercak – bercak kecil, kasar, dan berwarna coklat atau kehitaman pada daun (Meng et al., 2020).

Petani sangat perlu untuk menjaga kesehatan tanaman teh. Hal tersebut tidak mudah untuk dilakukan secara manual karena petani harus setiap saat datang ke kebun untuk memastikan bahwa tanaman tersebut tidak terkena penyakit.

Saat ini, diagnosis penyakit daun teh mengandalkan metode manual. Padahal, sebagian besar pohon teh tumbuh di daerah pegunungan yang terjal, sehingga metode manual ini memakan waktu dan biaya untuk melakukan diagnosis suatu penyakit teh. Untuk mengidentifikasi jenis penyakit daun teh secara akurat dan efisien sangat perlu dilakukan supaya meningkatkan kualitas teh dan meningkatkan pendapatan petani teh.

Dalam beberapa tahun terakhir, para peneliti telah menggunakan *deep learning*, khususnya *Convolutional Neural Network* (CNN), untuk mengidentifikasi hama dan penyakit dari tanaman. Dibandingkan dengan metode pembelajaran mesin klasik, metode *deep learning* dapat secara otomatis mengekstrak fitur penyakit tanaman dan memiliki akurasi yang lebih tinggi dalam mengidentifikasi penyakit tanaman (Bao et al., 2022).

Deteksi dan identifikasi penyakit daun teh yang akurat sangat dibutuhkan untuk pencegahan penyakit dan untuk mengendalikan penyakit dengan cara yang tepat. *Convolutional Neural Network* (CNN) dapat secara otomatis mengekstrak fitur daun teh yang sakit dalam gambar. Namun, gambar daun teh yang diambil di lingkungan memiliki masalah, seperti latar belakang yang rumit, daun yang lebat, dan perubahan skala besar. CNN memiliki akurasi yang rendah dalam mendekripsi dan mengi-

dentifikasi penyakit daun teh. Oleh karena itu pada penelitian ini mengusulkan untuk menggunakan metode arsitektur DenseNet untuk mendeteksi penyakit daun teh (Bao et al., 2022).

DenseNet memiliki beberapa kelebihan seperti, dapat mengatasi masalah *vanishing-gradient*, memperkuat propagasi fitur, mendorong penggunaan ulang fitur, dan secara substansial mengurangi jumlah parameter (Huang et al., 2017). Dengan metode tersebut diharapkan mampu mendapatkan model penyakit daun teh dengan akurasi yang cukup baik.

1.2 Rumusan Masalah

Rumusan masalah pada penelitian ini adalah bagaimana akurasi model klasifikasi penyakit daun teh yang baik dengan menggunakan arsitektur DenseNet-121.

1.3 Batasan Masalah

Pada penelitian yang dilakukan memiliki batasan masalah, yaitu

1. Pengambilan data dari Mendeley Data dan pengambilan data dilakukan dalam kurun waktu tertentu yang ditentukan.
2. Jumlah jenis penyakit daun teh dibatasi hanya tiga penyakit berdasarkan dari Mendeley Data yaitu Tea Leaf Blight, Tea Red Leaf Spot, dan Tea Red Scab.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah untuk mendapatkan model klasifikasi penyakit daun teh dengan akurasi model yang baik.

1.5 Manfaat Penelitian

1. Dari pemodelan yang telah dikerjakan dapat ditindaklanjuti dan digunakan kedalam kehidupan nyata seperti membuat drone khusus untuk mendeteksi jika ada penyakit pada tanaman.
2. Dengan adanya model klasifikasi tanaman daun teh, para petani teh diharapkan akan lebih mudah untuk mengidentifikasi jenis penyakit daun teh.





BAB II

TINJAUAN PUSTAKA

2.1 Penyakit Daun Teh

Teh sering terkena berbagai macam penyakit, seperti *Tea Leaf Blight*, *Tea Red Leaf Spot*, *Tea Red Scab*. Penyakit – penyakit ini mempengaruhi kualitas dan hasil Teh yang mengakibatkan kerugian ekonomi langsung bagi petani Teh (Meng et al., 2020).

Tea Leaf Blight, yaitu penyakit hawar daun Teh, penyakit teh yang umum di Tiongkok. Penyakit ini terdistribusikan di semua daerah penghasil teh di negara tersebut. Daun yang sakit sering gugur sebelum waktunya, dan pucuk baru menjadi layu ketika pohon Teh terinfeksi, sehingga mengakibatkan lemahnya kekuatan pohon. Penyakit *Tea Red Leaf Spot* adalah penyakit bercak pada daun yang biasanya berwarna coklat kemerah-merahan. Sedangkan *Tea Red Scab* merupakan penyakit yang memiliki bercak – bercak kecil, kasar, dan berwarna coklat atau kehitaman pada daun (Meng et al., 2020).

Klasifikasi penyakit Teh yang akurat dapat memfasilitasi tindakan pencegahan dan pengobatan pada tanaman tersebut. Hal ini tidak hanya mempromosikan industrialisasi pertanian, tetapi juga meningkatkan tingkat industri pertanian, dengan signifikan, praktis dan memiliki nilai sosial. Penelitian ini menggunakan pembelajaran mesin (*machine learning*). *Machine Learning* merancang cara kerja komputer dalam mengklasifikasi atau mendeteksi penyakit tanaman dengan menggunakan sebuah metode tertentu (Meng et al., 2020).

2.2 Pembelajaran Mesin (Machine Learning)

Pembelajaran mesin atau yang dikenal dengan sebutan *machine learning* adalah suatu bidang dalam *artificial intelligence* (AI), yang secara luas didefinisikan sebagai kemampuan mesin untuk meniru perilaku manusia yang cerdas. Sistem kecerdasan buatan digunakan untuk melakukan tugas-tugas kompleks dengan cara yang mirip dengan cara manusia memecahkan masalah (Brown, 2021).

Tujuan AI adalah untuk membuat model komputer yang menunjukkan “perilaku cerdas” seperti manusia, menurut Boris Katz, ilmuwan riset utama dan kepala Grup InfoLab di CSAI. Ini berarti mesin yang dapat mengenali pemandangan visual, memahami teks yang ditulis dalam bahasa alami, atau melakukan tindakan di dunia fisik. Pembelajaran mesin adalah salah satu cara untuk menggunakan AI. Hal ini didefinisikan pada 1950-an oleh pelopor AI, Arthur Samuel, sebagai “bidang studi yang memberi komputer kemampuan untuk belajar tanpa diprogram secara eksplisit.” (Brown, 2021).

Ada 3 subkategori pembelajaran mesin:

1. **Supervised machine learning models** adalah model pembelajaran mesin yang diawasi dilatih dengan kumpulan data berlabel, yang memungkinkan model untuk belajar dan tumbuh lebih akurat dari waktu ke waktu. Misalnya, algoritma akan dilatih dengan gambar anjing dan benda

lain, setiap gambar diberi label oleh manusia, lalu mesin akan mempelajari sendiri cara mengidentifikasi gambar anjing. Pembelajaran mesin yang diawasi adalah jenis yang paling umum digunakan saat ini (Brown, 2021).

2. ***Unsupervised machine learning***, yaitu pembelajaran mesin tanpa pengawasan. Pada *unsupervised machine learning*, sebuah program mencari pola dalam data yang tidak berlabel. Pembelajaran mesin tanpa pengawasan dapat menemukan pola atau tren yang tidak terlihat secara eksplisit. Misalnya, program pembelajaran mesin tanpa pengawasan dapat melihat data penjualan online dan mengidentifikasi berbagai jenis klien yang melakukan pembelian (Brown, 2021).
3. ***Reinforcement machine learning*** merupakan pembelajaran mesin penguatan, yaitu melatih mesin melalui coba – coba untuk mengambil tindakan terbaik dengan menetapkan sistem penghargaan. *Reinforcement learning* dapat melatih model untuk bermain game atau melatih kendaraan otonom untuk mengemudi dengan memberi tahu mesin saat membuat keputusan yang tepat, yang membantunya mempelajari tindakan apa yang harus diambil dari waktu ke waktu (Brown, 2021).

2.3 Convolutional Neural Network (CNN)

Convoultional networks, yang juga dikenal sebagai *Convolutional Neural Networks* (CNN) merupakan tipe jaringan saraf yang dirancang khusus untuk mengolah data dengan struktur mirip jaringan berbentuk kisi. Contohnya data deret waktu, yang dapat dianggap sebagai kisi 1-D, yang mengambil sampel pada interval waktu reguler, dan data gambar, yang dapat dianggap sebagai kisi piksel 2-D. Jaringan konvolusional sangat berhasil dalam aplikasi praktis. CNN menggambarkan penggunaan operasi matematika yang dikenal sebagai konvolusi. Konsep konvolusi merupakan bentuk khusus dari operasi linier. Jaringan konvolusional merujuk kepada jaringan saraf yang menerapkan konvolusi sebagai alternatif dari perkalian matriks biasa pada setidaknya satu lapisannya (Goodfellow et al., 2016).

Dua lapisan awal dalam CNN, yaitu lapisan konvolusi dan lapisan penggabungan, berperan dalam mengambil ciri-ciri dari data, sementara lapisan ketiga yang sepenuhnya terhubung bertugas memetakan ciri-ciri yang telah diambil menjadi hasil akhir, seperti dalam tugas klasifikasi. Lapisan konvolusi memiliki peranan sentral dalam struktur CNN, terdiri dari rangkaian operasi matematis, seperti konvolusi, yang merupakan bentuk operasi linier khusus. Dalam konteks gambar digital, nilai piksel disusun dalam kisi dua dimensi (2D), yang mirip dengan susunan angka pada Gambar 2.1. Kisi kecil yang dikenal sebagai kernel, bertindak sebagai alat untuk mengekstrak fitur yang dapat dioptimalkan. Kernel ini diterapkan pada setiap posisi dalam gambar, menjadikan CNN sangat efisien untuk mengolah gambar karena fitur-fitur penting dapat muncul di berbagai lokasi pada gambar. Ketika satu lapisan mengirimkan hasilnya ke lapisan berikutnya, fitur-fitur yang telah diekstrak secara bertahap menjadi lebih kompleks secara hierarkis. Proses penyesuaian parameter seperti kernel disebut pelatihan. Hal ini dilakukan untuk meminimalkan perbedaan antara hasil keluaran dan label acuan yang benar melalui algoritma yang dikenal sebagai propagasi balik (*backpropagation*) dan penurunan gradien (*gradient descent*). (Yamashita et al., 2018).



0	2	15	0	0	11	10	0	0	0	0	9	9	0	0	0	0
0	0	0	4	60	157	236	255	255	177	95	61	32	0	0	29	
0	10	16	119	238	255	244	245	243	250	249	255	222	103	10	0	
0	14	170	255	255	244	254	255	253	245	255	249	253	251	124	1	
2	98	255	228	255	251	254	211	141	116	122	215	251	238	255	49	
13	217	243	255	155	33	226	52	2	0	10	13	232	255	255	36	
16	229	252	254	49	12	0	7	7	0	70	237	252	235	62		
6	141	245	255	212	25	11	9	3	0	115	236	243	255	137	0	
0	87	252	250	248	215	60	0	112	252	255	248	144	6	0		
0	13	113	255	255	245	255	182	181	248	252	242	208	36	0	19	
1	0	5	117	251	255	241	255	247	255	241	162	17	0	7	0	
0	0	0	4	58	251	255	246	254	253	255	120	11	0	1	0	
0	0	4	97	255	255	255	248	252	255	244	255	182	10	0	4	
0	22	206	252	246	251	241	100	24	118	255	245	255	194	9	0	
0	0	111	255	242	242	255	158	24	0	0	6	39	255	232	230	
0	218	251	250	137	7	11	0	0	0	2	62	255	250	125	3	
0	173	255	255	101	9	20	0	13	3	13	182	251	245	61	0	
0	107	251	241	255	230	98	55	19	118	217	248	253	255	52	4	
0	18	146	250	255	247	255	255	249	255	240	255	129	0	5		
0	0	23	113	215	255	250	248	255	255	248	248	118	14	12	0	
0	0	6	1	0	52	153	233	255	252	147	37	0	0	4	1	
0	0	5	5	0	0	0	0	0	14	1	0	6	6	0	0	

Gambar 2.1. Array of numbers (Yamashita et al., 2018)

Sistem komputer menginterpretasikan gambar sebagai rangkaian nilai numerik. Di sebelah kanan, terdapat suatu matriks yang berisi angka dalam rentang nol (0) hingga 255, di mana tiap angka mencerminkan tingkat kecerahan piksel pada gambar yang tampak di sebelah kiri. Kedua representasi ini ditampilkan dalam gambar di tengah.

2.4 Membangun Blok Arsitektur CNN

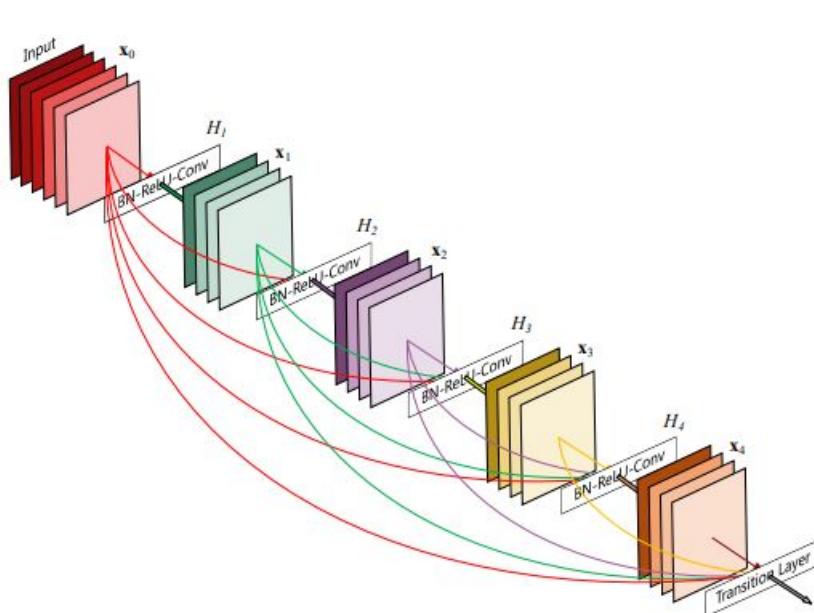
Arsitektur CNN mencakup beberapa blok bangunan, yaitu lapisan konvolusi, lapisan penyatuan, dan lapisan yang terhubung sepenuhnya. Arsitektur CNN terdiri dari pengulangan tumpukan beberapa lapisan konvolusi (*convolution layers*) dan lapisan penyatuan (*pooling layers*), diikuti oleh satu atau lebih lapisan yang terhubung sepenuhnya. Langkah di mana data masukan diubah menjadi keluaran melalui lapisan – lapisan ini disebut perambatan maju (*forward propagation*). Meskipun operasi konvolusi dan penyatuan yang dijelaskan di bagian ini adalah untuk 2D-CNN, operasi serupa juga dapat dilakukan untuk tiga dimensi (3D)-CNN. Lapisan konvolusi adalah komponen mendasar dari arsitektur CNN yang melakukan ekstraksi fitur, yang biasanya terdiri dari kombinasi operasi linier dan non-linier, yaitu operasi konvolusi dan fungsi aktivasi (Yamashita et al., 2018).

2.5 DenseNet

Klasifikasi gambar adalah tugas visi komputer yang mendasar dan penting. Dalam beberapa tahun terakhir, *convolutional neural network* (CNN) telah menjadi pendekatan pembelajaran mesin yang dominan untuk pengenalan visi komputer. CNN yang sebelumnya banyak digunakan adalah *Original LeNet5* dan *ResNet*. Kedua model ini memiliki beberapa potensi masalah seperti terlalu banyak parameter, hilangnya gradien, dan kesulitan dalam pelatihan (Zhong et al., 2020).

Dense Convolutional Network (DenseNet) memiliki konektivitas yang padat dibandingkan dengan model CNN lain seperti *Vgg* dan *Resnet*. DenseNet dapat mengatasi masalah gradien menghilang, meningkatkan penyebaran peta fitur, dan mengurangi jumlah parameter. Pada model DenseNet, pola konektivitas yang berbeda dengan CNN lainnya adalah koneksi langsung dari setiap lapisan ke semua lapisan berikutnya, yang selanjutnya dapat meningkatkan arus informasi antar lapisan (Zhong et al., 2020).

DenseNet adalah arsitektur modern CNN untuk pengenalan objek visual yang telah memperoleh *state-of-the-art* dengan parameter yang lebih sedikit. Dengan beberapa modifikasi utama, DenseNet sangat mirip dengan ResNet. DenseNet, bersama dengan atribut *concatenate* (.), menggabungkan keluaran lapisan sebelumnya dengan lapisan yang akan datang, sementara ResNet menggunakan atribut tambahan (+) untuk menggunakan lapisan sebelumnya dengan lapisan yang akan datang. Arsitektur DenseNet bertujuan untuk mengatasi masalah ini dengan menghubungkan semua lapisan secara padat. Umumnya, CNN tradisional menghitung lapisan keluaran (ke- l) menggunakan transformasi non-linier H_l (.) ke keluaran lapisan sebelumnya \mathbf{X}_{l-1} (Zhong et al., 2020).



Gambar 2.2. Alur Arsitektur DenseNet (Huang et al., 2017)

$$\mathbf{X}_l = H_l(\mathbf{X}_{l-1})$$

DenseNet menawarkan model komunikasi yang mudah untuk meningkatkan arus informasi antar lapisan: lapisan ke-1 menerima masukan dari fitur semua tingkat sebelumnya. Persamaan ini kemudian diubah lagi menjadi

$$\mathbf{X}_l = H_l[\mathbf{X}_0, \mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_{l-1}]$$

Dimana $\mathbf{X}_l = H_l[\mathbf{X}_0, \mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_{l-1}]$ adalah tensor tunggal yang dibentuk oleh rangkaian peta keluaran dari lapisan sebelumnya. Di luar dari fungsinya, $H_l(\cdot)$ mewakili fungsi transformasi non-linier. Fungsi ini terdiri dari tiga operasi utama, *batch normalization* (BN), *activation* (ReLU) dan *pooling* dan konvolusi (CONV) (Hasan et al., 2021).

Arsitektur DenseNet telah dipresentasikan pada Gambar 2.2. Bagaimanapun juga, peningkatan rate k membantu untuk menggeneralisasikan lapisan ke- l yang mana: $k^{[l]} = (l^{[0]} + k^{(l-1)})$, di mana $k^{[0]}$ diketahui sebagai angka dari dimensi terakhir (Hasan et al., 2021).

1. Dilarang mengutip karya tulis ini, kecuali:
 - a. menyebutkan sumber sesuai kaidah kecendekiaan
 - b. pengutipan hanya untuk keperluan pendidikan, penulisan satya ilmiah atau referensi
 - c. pengutipan tidak merugikan Universitas Pertamina.
2. Dilarang mempublikasikan atau memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa adanya izin dari Universitas Pertamina.

- ## 2.6 Laporan Klasifikasi
- Laporan klasifikasi adalah metode yang digunakan dalam *machine learning* untuk mengevaluasi performa model klasifikasi. Ini memberikan ringkasan komprehensif dari berbagai metrik yang menilai keakuratan dan efektivitas prediksi model. Berikut penjelasan komponen utama yang biasanya disertakan dalam laporan klasifikasi (Chouinard, 2022).
1. *Precision*: Presisi mengukur proporsi kejadian positif yang diprediksi dengan tepat dari seluruh kejadian yang diprediksi positif. *Precision* dihitung sebagai rasio positif benar terhadap jumlah positif benar dan positif palsu. Presisi yang lebih tinggi menunjukkan lebih sedikit kesalahan positif (Annamalai, 2018).
 2. *Recall*: *Recall* juga dikenal sebagai sensitivitas atau tingkat positif sebenarnya, mengukur proporsi kejadian positif yang diprediksi dengan tepat dari semua kejadian positif aktual. *Recall* dihitung sebagai rasio positif benar terhadap jumlah positif benar dan negatif palsu (Chouinard, 2022).
 3. *F1-score*: *F1-score* adalah rata - rata harmonik antara presisi dan *recall*. *F1-score* memberikan ukuran kinerja model yang seimbang dengan mempertimbangkan presisi dan *recall*. *F1-score* dihitung sebagai 2 kali produk presisi dan *recall* dibagi dengan jumlah presisi dan *recall* (Chouinard, 2022).
 4. *Support*: *Support* mengacu pada jumlah kemunculan setiap kelas dalam kumpulan data *test*. *Support* memberikan wawasan tentang distribusi kelas dan dapat membantu mengidentifikasi kumpulan data yang tidak seimbang (Chouinard, 2022).
 5. *Macro Avg*: *Macro-averaged score* dihitung dengan mengambil rata - rata aritmatika dari skor individu setiap label kelas. *Macro-averaged score* memperlakukan semua kelas secara merata dan memberikan bobot yang sama pada setiap kelas saat menghitung rata - rata. Metode ini berguna ketika semua kelas perlu dievaluasi secara merata untuk menilai kinerja pengklasifikasi secara keseluruhan sehubungan dengan label kelas (Masoud, 2022).
 6. *Weighted Avg*: *Weighted-averaged* dihitung dengan memberi bobot pada skor setiap label kelas dengan jumlah kejadian sebenarnya saat menghitung rata-rata. Hal ini berguna dalam kasus ketidakseimbangan kelas di mana jumlah instance yang berbeda dikaitkan dengan label kelas yang berbeda. Metode ini memberikan bobot lebih pada kelas dengan lebih banyak *instance*, yang dapat membantu menyeimbangkan dampak data yang tidak seimbang (Masoud, 2022).

2.7 Penelitian Terkait

Swaminathan et al. (2021) melakukan penelitian yang terkait dengan tanaman – tanaman menggunakan DenseNet-121 dengan judul “MULTIPLE PLANT LEAF DISEASE CLASSIFICATION USING DENSENET-121 ARCHITECTURE”. Pada penelitiannya, mereka mencari tahu atau mendekripsi penyakit dari tanaman apel, ceri, jagung, anggur, paprika, kentang, dan tomat. Akurasi rata – rata dari *feature set based multiple disease classifier* (FSMDC) adalah 93,2%. Rata – rata keseluruhan

yang mereka dapatkan adalah 98,23% yang mengungguli metode yang diusulkan lainnya. Pada artikel mereka dijelaskan bahwa rata – rata teoritis ini dihitung dengan meratakan akurasi masing – masing tanaman dalam mendeteksi kelas masing – masing. Juga secara praktis yaitu menjalankan semua 29 label dari kelas tumbuhan berbeda yang diberikan pada model untuk klasifikasi, mereka mencapai akurasi 94,96%.

Saleem et al. (2020) melakukan penelitian terkait penyakit pada tanaman juga. Mereka melakukan perbandingan metode arsitektur CNN yang terdiri dari banyak arsitektur, salah satunya menggunakan DenseNet-121. Pada penelitiannya, dataset terdiri dari total 54.306 gambar dengan label 38 daun sehat/berpenyakit yang berbeda terkait dengan 14 spesies tanamannya. Pada arsitektur yang menggunakan DenseNet, mereka mendapatkan nilai *Training Accuracy* sebesar 0.9826, dan nilai *Validation Accuracy* 0.9580.

Dari kedua penelitian tersebut, metode yang digunakan adalah DenseNet-121 yang diimplementasikan pada model klasifikasi penyakit tanaman. Akurasi dari kedua model tersebut sangat bagus yaitu 94,96% untuk data *test* (Swaminathan et al., 2021), 98,26% untuk data *train* dan 95,8% untuk data validasi (Saleem et al., 2020).

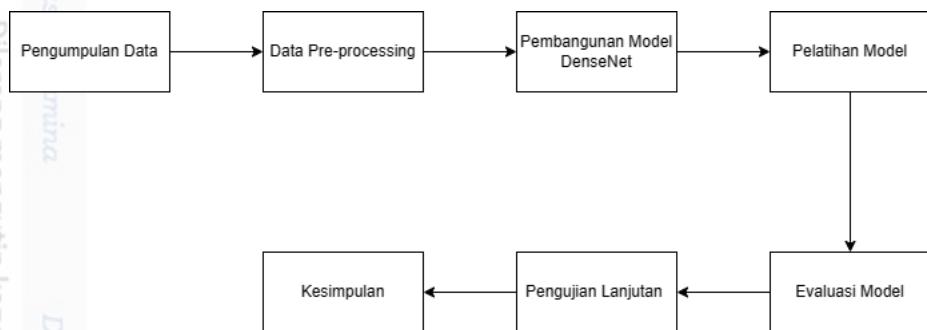
Dilindungi Undang-Undang
© Copyright of Universitas Pertamina
Karya tulis ini, kecuali:
1. Penggunaan sumber sesuai kaidah kecendekian;
2. Dilarang mempublikasikan atau memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun
tanpa adanya izin dari Universitas Pertamina.





3.1 Alur Penelitian

Pada penelitian ini, tahapan yang dilakukan mengikuti alur pada Gambar 3.1, secara detail penjelasan dari gambar tersebut disajikan pada sub-bab berikutnya pada bab ini.



Gambar 3.1. Alur Proses Klasifikasi Gambar

3.2 Pengumpulan Data

Dalam prosedur pengumpulan data, penelitian ini memperoleh data gambar daun teh yang mana terdiri dari 3 penyakit, yaitu *Tea Red Leaf Spot*, *Tea Leaf Blight*, dan *Tea Red Scab*. Data ini telah diunggah pada tahun 2019 oleh Gensheng Hu, Haoyu Wu, Yan Zhang, dan Mingzhu Wan dengan judul Data for: A Low Shot Learning Method for Tea Leaf's Disease Identification. Data ini terdiri dari data *train* sebanyak 240 gambar untuk masing - masing kelas, kemudian data *test* terdiri 20 gambar untuk masing - masing kelas (Hu et al., 2019). Ukuran data gambar tersebut tidak sama, di antaranya memiliki ukuran 188×308 , 233×160 , 269×187 , dan berbagai ukuran lainnya.

3.3 Pra-pemrosesan Data

Data yang telah dikumpulkan sudah dalam bentuk *train* dan *test* yang mana terdiri dari 3 kelas, yaitu *Tea Red Leaf Spot*, *Tea Leaf Blight*, dan *Tea Red Scab*. Pada tahap ini, data *train* dan data *test* diinput dan dilakukan pra-pemrosesan data yang mana data gambar dari data *train* dan juga *test* diseragamkan ukurannya menjadi 224×224 , kemudian ukuran *batch* diseragamkan menjadi 16. Data tersebut dipanggil menggunakan metode *tensorflow*. Pengubahan ukuran gambar dilakukan supaya program dapat mudah mengenali data, dan selanjutnya dataset yang digunakan pada penelitian ini relatif sedikit, sehingga ukuran *batch* dibuat menjadi 16.

Sebelum masuk ke tahap pemodelan, dilakukan *resize* dan *rescalling* dan augmentasi data. *Resizing* dilakukan untuk mengubah ukuran menjadi 224×224 piksel dan mengubah skala gambar dengan faktor $1.0/255$. Tujuannya adalah untuk mengubah intensitas piksel dari rentang nol (0) sampai 255 menjadi rentang nol (0) sampai satu (1). Hal ini dilakukan karena banyak model pembelajaran

mesin yang lebih mudah untuk dilatih ketika data berada dalam rentang yang lebih kecil seperti nol (0) sampai satu (1). Augmentasi data adalah teknik yang umum digunakan dalam pelatihan model machine learning, terutama dalam tugas *computer vision*, untuk meningkatkan variasi data pelatihan dan membantu mengurangi *overfitting*.

Dari dataset yang telah dikumpulkan perlu dipisahkan menjadi data *train* dan data validasi. Data validasi digunakan untuk mengevaluasi kinerja model selama proses pelatihan. Hal ini membantu untuk melihat seberapa baik model dapat melakukan generalisasi pada data yang belum pernah dilihat sebelumnya. Validasi set berfungsi untuk mengestimasi kinerja model pada data yang tidak terlihat dan membantu menghindari *overfitting* (terlalu beradaptasi dengan data pelatihan) atau *underfitting* (tidak cukup belajar dari data pelatihan). Data *train* diambil sebanyak 80% dan data *validation* diambil sebanyak 20% dengan metode 'tf.dataset' yaitu menggunakan data *train* '.take(36)', dan data *validation* '.skip(36)', yang artinya data *train* diambil sebanyak 36 *batch* dan data *validation* sisanya dari totalnya 45-36, yaitu 9 *batch*.

3.4 Pembangunan Model DenseNet

Setelah data dilakukan pra-pemrosesan, dibuatlah arsitektur DenseNet. Penelitian ini menggunakan DenseNet-121 dengan juga penyesuaian parameter seperti jumlah blok Dense, jumlah filter, dan ukuran kernel sesuai dengan data dan kompleksitas masalah yang ingin diselesaikan dengan beberapa kali percobaan untuk mendapatkan akurasi yang baik.

3.5 Pelatihan Model

Pada bagian ini ditentukan fungsi *loss* yang sesuai yaitu *sparse categorical cross-entropy*. Model dilatih menggunakan data pelatihan (*train*), lalu dimunculkan hasil akurasi dan *loss* pada set validasi selama pelatihan. Lalu juga menggunakan teknik regularisasi seperti *dropout* atau *weight decay* selanjutnya digunakan untuk mencegah *overfitting*.

3.6 Evaluasi Model

Evaluasi model dilakukan pada set pengujian (*test*) yang tidak pernah dilihat, sebelumnya untuk mendapatkan perkiraan kinerja yang objektif. Selanjutnya dihitung metrik evaluasi seperti akurasi, presisi, *recall*, dan *F1-score*. Metrik evaluasi ini dianalisa untuk mengevaluasi performa model dalam mengklasifikasikan penyakit daun teh.

3.7 Pengujian Lanjutan

Tahap pengujian lanjutan ini dilakukan ketika model belum mencapai performa yang memuaskan. Beberapa cara untuk meningkatkan dan mengoptimalkan kinerja model DenseNet di antaranya, optimasi hyperparameter, penyesuaian arsitektur, dan regularisasi untuk mencegah *overfitting* pada model. Pada tahap regularisasi digunakan *dropout* untuk mengurangi kompleksitas model dan meningkatkan generalisasi pada data pengujian.

Augmentasi data digunakan untuk meningkatkan variasi data pelatihan dan mengurangi risiko *overfitting* dengan menerapkan rotasi, dan *flip* horizontal/vertikal. Selanjutnya, evaluasi dan analisis eror terhadap hasil prediksi model pada data pengujian dan analisis kesalahan yang dilakukan. Hal ini

dilakukan untuk mengidentifikasi pola kesalahan yang sering terjadi dan mengambil tindakan untuk memperbaiki model, seperti memperbaiki fitur atau label yang ambigu.



1. Dilarang mengutip karya tulis ini, kecuali:

- a. menyebutkan sumber sesuai kaidah kecendekian;
- b. pengutipan hanya untuk keperluan pendidikan, penulisan karya ilmiah atau penelitian;
- c. pengutipan tidak merugikan Universitas Pertamina.

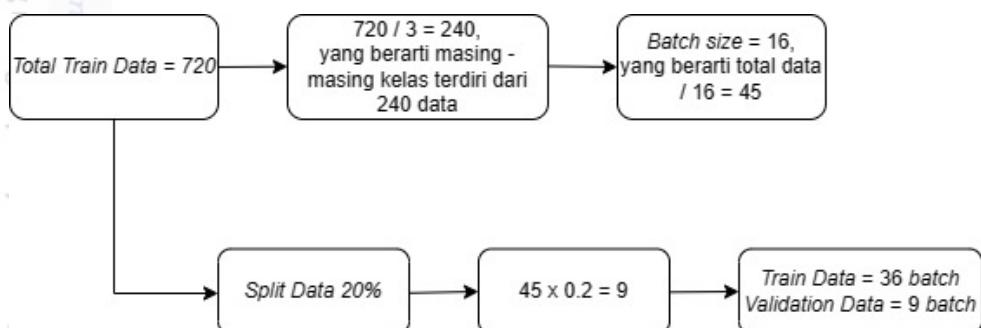
2. Dilarang mempublikasikan atau memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa adanya izin dari Universitas Pertamina.





4.1 Proses Pengolahan Data

Pada penelitian ini, telah dilakukan tahapan pemodelan arsitektur DenseNet-121 yang digunakan untuk mengklasifikasi penyakit daun teh dari dataset yang telah dikumpulkan. Sebelum masuk ke pemodelan, langkah awal yang dilakukan adalah memasukkan *library* yang dibutuhkan dan membaca data. Selanjutnya dilakukan *pre-processing* data gambar dan pemisahan validasi. Pada proses ini gambar diolah dengan mengubah ukuran *batch* menjadi 16, dan ukuran gambar sebesar 224×224 .



Gambar 4.1. Ilustrasi Pembagian Data

Pada Gambar 4.1 terlihat ilustrasi pemisahan antara data latih dan data validasi yang dilakukan dengan cara menghitung jumlah *batch* dari data *train*, di mana data *train* sebanyak 720 data dibagi menjadi 45 *batch* data yang berarti 1 *batch* terdiri dari 16 data.

Sebelum masuk ke tahap pemodelan, dipersiapkan variabel *resize scaling* dan *augmentation data*. *Resize scaling* dilakukan untuk mengubah ukuran gambar, namun dikarenakan sebelumnya telah memasukkan data dengan parameter '*image_size=(224, 224)*', maka tidak perlu untuk mengubah ukuran (*resize*) lagi. Kemudian, skala gambar diubah dengan faktor $1.0/255$. Hal ini bertujuan untuk mengubah intensitas piksel dari rentang nol (0) sampai 255 menjadi rentang nol (0) sampai satu (1). Hal ini dilakukan karena DenseNet-121 lebih mudah untuk dilatih ketika data berada dalam rentang yang lebih kecil seperti nol (0) sampai satu (1). Variabel augmentasi data digunakan untuk meningkatkan variasi data latih dan membantu mengurangi *overfitting*.

Tahap selanjutnya adalah pemodelan arsitektur DenseNet-121. Pada arsitektur DenseNet, di setiap *dense block* terdapat tahapan *Batch Normalization* (BN) - *ReLU* - konvolusi (CONV). Pada DenseNet-121 terjadi proses awal yaitu konvolusi dengan kernel 7×7 , *stride* 2, lalu *pooling layer* dengan *pool size* 3×3 , *stride* 2. Setelah itu, dilakukan tahap *Dense Block*. Pada tahap ini berjalan proses '*bn_relu_conv*' yang dimulai dengan jumlah kernel 1×1 dan dilanjutkan dengan kernel 3×3 . Tahap *Dense Block* dilakukan sebanyak empat kali dengan masing-masing proses *Dense Block* memiliki pengulangan yang berbeda, yaitu

1. *Dense Block* (1) dilakukan sebanyak enam (6) kali, yang berarti memiliki 6 lapisan konvolusi,
2. *Dense Block* (2) 12 kali, memiliki 12 lapisan konvolusi
3. *Dense Block* (3) 24 kali, memiliki 24 lapisan konvolusi, dan
4. *Dense Block* (4) sebanyak 16 kali, memiliki 16 lapisan konvolusi,

Diantara proses *Dense Block* terdapat proses '*Transition layer*'. Tahap '*Transition layer*' dilakukan proses 'bn_relu_conv' dengan kernel 1×1 dan diikuti dengan proses *average pool* dengan *pool size* 2×2 , *stride* 2. Proses ini dilakukan sebanyak tiga kali. Kemudian untuk *Dense Block* yang ke empat tidak dilanjutkan dengan proses *Transition Layer*, melainkan masuk ke proses *Classification Layer*, yang terjadi di proses ini yaitu *Global Average Pool* yang diikuti dengan proses *Dropout*.

Dropout digunakan untuk mencegah *overfitting* dengan mengaktifkan atau menonaktifkan neuron secara acak pada setiap iterasi pelatihan. Probabilitas *dropout* sebesar 0,5 berarti setiap neuron memiliki peluang 50% untuk dinonaktifkan. Ini membantu mencegah jaringan untuk terlalu bergantung pada setiap neuron tertentu. Setelah *dropout*, dilakukan operasi *dense* yang memiliki 1.024 neuron dengan fungsi aktivasi '*ReLU*', lalu dilakukan operasi *dense* lagi dengan 512 neuron. Untuk menjaga distribusi nilai dalam lapisan *dense* yang baru saja ditambahkan, digunakan *Batch Normalization*. Lalu dilakukan proses *dropout* lagi untuk mencegah *overfitting* di lapisan *dense* tambahan.

Tahap terakhir adalah tahap membangun layer yang sepenuhnya terhubung (*fully connected*). Pada tensorflow proses kode ini dipanggil dengan '**tf.keras.layers.Dense**' dengan parameter **n_classes** sebagai jumlah kelas, dan **activation='softmax'**'. Parameter *activation* diisi dengan *softmax* karena data yang telah dikumpulkan memiliki lebih dari dua (2) kelas.

Tahap berikutnya adalah tahap kompilasi yang menggunakan *optimizer Adam (Adaptive Moment Estimation)* dengan *learning rate* sebesar 0.0001. *Learning rate* adalah parameter yang mengontrol seberapa besar langkah perubahan yang diambil oleh algoritma optimasi untuk mencari nilai-nilai parameter model yang optimal. Dalam setiap *learning state* digunakan fungsi *loss Sparse Categorical Crossentropy*. Fungsi ini digunakan untuk mengatasi masalah klasifikasi dengan beberapa kelas dan label yang diwakili oleh angka-angka bukan *one-hot encoded*. Dalam konteks ini model diharapkan untuk memberikan *output* kelas yang merupakan angka yang sesuai dengan kelas yang benar. Fungsi *loss* ini menghitung perbedaan antara prediksi dan label sesuai dengan model yang dikompilasi.

Parameter *run_eagerly=True* diisi *true* untuk memastikan dapat dilakukannya *debugging* dan pengembangan, karena memungkinkan untuk melihat hasil dari setiap operasi dengan cepat dan tidak memerlukan konstruksi graf terpisah. Parameter *metrics* merupakan parameter untuk menentukan metrik evaluasi yang ingin dipantau selama pelatihan dan evaluasi model. Parameter ini diisi *accuracy* karena akurasi dipilih sebagai metrik evaluasi. Akurasi adalah metrik yang umum digunakan untuk masalah klasifikasi dan mengukur sejauh mana model dapat mengklasifikasi data dengan benar.

Setelah melakukan kompilasi model, selanjutnya dilakukan pelatihan model dengan memanggil metode '*fit*'. Parameter yang digunakan adalah data *train*, '*epoch=50*', '*batch_size=16*', '*verbose=2*', '*callbacks=[anne, checkpoint]*', dan data validasi. *Verbose* adalah tingkat keterangan yang akan di-

gunakan selama pelatihan. Nilai '*verbose=2*' akan menampilkan satu bar per *epoch* dan akan menunjukkan progres latihan.

Pada '*callbacks*', terdapat varibel '*anne*' dan '*checkpoint*'. '*Anne*' adalah *callback* '*ReduceLROnPlateau*' yang merupakan salah satu *callback* bawaan dari *TensorFlow* yang digunakan untuk mengatur penyesuaian pembelajaran secara otomatis berdasarkan metrik yang dipantau. '*ReduceLROnPlateau*' adalah nama dari objek *callback* yang digunakan untuk mengurangi pembelajaran pada '*plateau*' (daerah di mana performa model tidak mengalami peningkatan yang signifikan) pada data validasi. Di dalam *callback* ini terdapat juga parameter seperti '*monitor=val_loss*' yaitu metrik yang dipantau untuk menentukan apakah performa model telah mencapai '*plateau*'. Disini hal yang dipantau adalah perubahan '*val_loss*' (loss pada data validasi). Pengisian parameter '*factor=0.3*' berarti faktor penurunan pembelajaran sebesar 0.3. Jika metrik yang dipantau telah mencapai "*plateau*", maka pembelajaran akan dikurangi dengan faktor ini dikalikan dengan 0.3. Parameter '*patience=3*' adalah jumlah *epoch* yang harus dilewati tanpa adanya peningkatan dalam metrik yang dipantau sebelum *callback* memutuskan untuk mengurangi pembelajaran. Dalam proses ini, jika '*val_loss*' tidak berkurang selama 3 *epoch* berturut-turut, *callback* akan diaktifkan dan pembelajaran akan dikurangi. Nilai '*verbose=2*' akan menampilkan satu bar per *epoch* dan akan menunjukkan progres latihan. Parameter '*min_lr=0.0003*' adalah pembatas terendah untuk pembelajaran. Jika pembelajaran telah dikurangi dan mencapai nilai '*min_lr*', maka pembelajaran tidak akan dikurangi lebih lanjut. Ini berfungsi untuk mencegah pembelajaran terlalu rendah.

Parameter '*ModelCheckpoint*' merupakan salah satu *callback* bawaan dari *TensorFlow* yang digunakan untuk menyimpan model pada titik tertentu selama pelatihan. Terdapat beberapa parameter lainnya seperti '*save_best_only*', ini adalah parameter yang menentukan apakah hanya model dengan performa terbaik yang akan disimpan. Karena diatur menjadi '*True*', '*callback*' akan menyimpan hanya model dengan performa terbaik (loss terendah) pada data validasi. Artinya, jika ada model yang memiliki performa lebih buruk dibandingkan dengan model sebelumnya, model tersebut tidak akan disimpan. Parameter '*mode=min*' adalah mode untuk memahami perubahan performa model. Karena di sini menggunakan '*mode=min*', berarti model disimpan ketika metrik yang dipantau memiliki nilai minimum yang lebih rendah.

Setelah melakukan pelatihan, diperoleh metrik *loss*, *accuracy*, *val_loss*, *val_accuracy*. Selanjutnya dilakukan proses evaluasi performa model pada data *test*. Proses evaluasi ini bertujuan untuk mengukur seberapa baik model dapat menggeneralisasi pada data yang belum pernah dilihat sebelumnya.

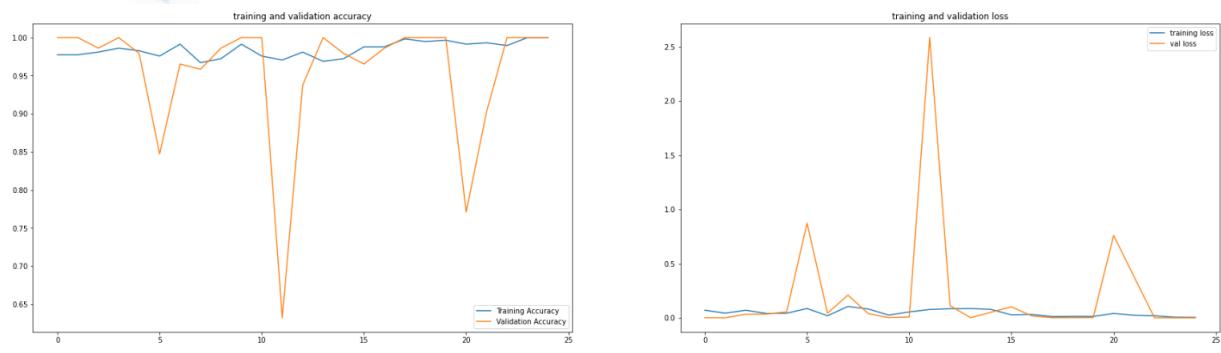
Pada penelitian ini, untuk melihat akurasi dari model dilakukan percobaan yaitu melakukan simulasi dengan mengubah nilai *epoch*. Berdasarkan penelitian terkait (Swaminathan et al., 2021), yaitu model klasifikasi penyakit tanaman yang dilihat dari daunnya dengan menggunakan metode yang sama, pada penelitian ini digunakan *epoch* 50. Selanjutnya pada penelitian Saleem et al. (2020) yang juga membahas model klasifikasi penyakit tanaman dengan membandingkan beberapa metode, digunakan *epoch* 60. Nilai *epoch* yang digunakan Saleem et al. (2020) mirip dengan yang digunakan Swaminathan et al. (2021). Oleh karena itu, pada penelitian ini digunakan nilai *epoch* 50. Pada penelitian ini digunakan pula nilai *epoch* 100, dan 300, untuk melihat pengaruh *epoch* terhadap

Universitas Pertamina

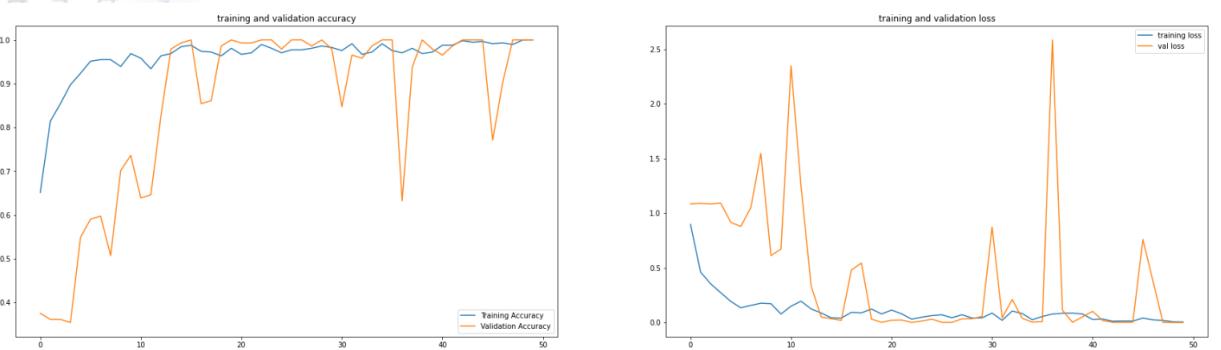
4.2 Hasil dan Pembahasan

Pada bagian ini akan dibahas hasil percobaan pada masing-masing epoch yang sudah ditentukan.

4.2.1 Percobaan Epoch = 50



Gambar 4.2. Evaluasi Score (25)



Gambar 4.3. Evaluasi Score (50)

Pada Gambar 4.2 terlihat nilai akurasi dan *loss* dari data *train* dan validasi sampai *epoch* 25 di mana nilai akurasi dan *loss* stabil pada saat *epoch* lebih dari 10 sampai dengan *epoch* 20. Pada Gambar 4.3 terlihat bahwa nilai akurasi dan *loss* dari data *train* dan validasi sampai *epoch* 50 di mana nilai akurasi dan *loss* stabil pada saat *epoch* antara 20 sampai dengan *epoch* 30.

Tabel 4.1. Evaluasi Score

Test Accuracy	0,933
Test Loss	0,351

Kemudian setelah melakukan evaluasi, dilakukan proses menampilkan perubahan akurasi dan *loss* pada data *training* dan validasi selama proses pelatihan. Pada gambar di atas dapat dilihat perubahan nilai akurasi dan *loss* dari data *train* dan validasi. Pada Tabel 4.1 diperlihatkan nilai akurasi yaitu 0.933 dan *loss* yaitu 0.351.

```

actual: red_scab
1/1 [=====] - 0s 129ms/step
predicted: red_scab

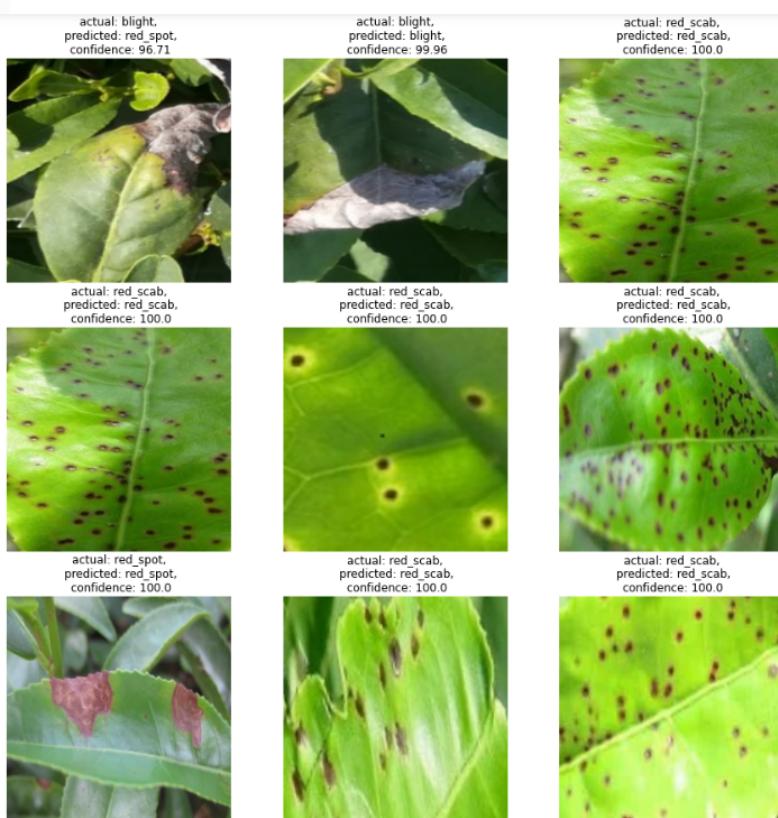
```



Gambar 4.4. Prediksi Label Pertama

Setelah melihat grafik tersebut, dilakukan percobaan awal yaitu dengan mengambil 1 gambar pertama apakah model yang telah dilatih dapat memprediksi gambar sesuai dengan data. Akhirnya mendapatkan hasil yang sesuai dengan ekspektasi di mana gambar teh dengan label penyakit *blight* sesuai diprediksi oleh model.

Selain melihat 1 gambar pertama, dibuatlah juga fungsi untuk memprediksi 9 gambar apakah sesuai dengan data *test*, yang mana pada fungsi tersebut mengeluarkan hasil kelas yang diprediksi dan tingkat kepercayaannya (*confidence*). Hasilnya dapat dilihat pada gambar berikut.



Gambar 4.5. Prediksi 9 Gambar

1. Dilarang ~~menyalin dan menyebarkan~~, kecuali:

 - a. menyalin dan menyebarkan untuk keperluan pendidikan, penulisan karya ilmiah atau penelitian;
 - b. pengutipan tidak merugikan Universitas Pertamina.
 - c. pengutipan tidak merugikan Universitas Pertamina.

Dari gambar 4.5 terlihat bahwa model dapat memprediksi 9 gambar dengan sangat baik dan benar.

Untuk melihat informasi seperti apa model dapat memprediksi data *test*, terlebih dahulu akan dilakukan yaitu pendefinisikan *model.predict* dalam sebuah variabel. Ini adalah perintah untuk memprediksi output dari data *test*. Hasil prediksi akan disimpan dalam sebuah variabel. Variabel yang telah disimpan berupa angka yang merupakan hasil dari *activation = softmax*. Angka tersebut adalah distribusi probabilitas terhadap data. Untuk menentukan ketepatan prediksi, perlu dibuat label prediksi dan label aktual. Langkah awal untuk mendapatkan label prediksi dan label aktual adalah dengan membuat list kosong dan kemudian dijalankan dengan iterasi melalui data *test* menggunakan *batch* dan mengumpulkan hasil prediksi dan label aktual dari model yang telah dilatih sebelumnya.

Penentuan label prediksi dan aktual juga diperlukan untuk mendapatkan array gambar dari data *test* yang dipakai dengan cara inisialisasi list kosong. Array gambar akan ditambahkan ke dalam variabel tersebut dengan fungsi '*extend*' untuk menggabungkan array dari *batch-batch* sebelumnya. Lalu array gambar tersebut dikonversi menjadi integer.

Langkah selanjutnya adalah melihat bagaimana hasil dari laporan klasifikasi yang membandingkan label aktual dan label prediksi dari model untuk data *test*. Hasil keluaran dari laporan klasifikasi mencakup beberapa metrik evaluasi seperti akurasi, *precision*, *recall*, dan *f1-score* untuk setiap kelas serta rata - rata dari metrik - metrik tersebut untuk seluruh kelas. Laporan klasifikasi ini sangat berguna untuk mengukur kinerja model secara lebih rinci dan memahami seberapa baik model dapat mengklasifikasikan data *test* ke dalam kelas yang benar. Dapat dilihat informasi yang diberikan yaitu rata - rata akurasinya 0.93 (93%), beserta *f1-score* terhadap masing - masing kelas. *F1-score* untuk kelas 0 (pertama:*blight*) adalah 0.95 (95%), kelas 1 (kedua:*red scab*) 0.95 (95%), dan kelas 2 (ketiga:*red spot*) 0.90 (90%). *F1-score* digunakan untuk mengukur kinerja model.

Tabel 4.2. Laporan Evaluasi

	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>	<i>Support</i>
0	1.00	0.90	0.95	20
1	0.91	1.00	0.95	20
2	0.90	0.90	0.90	20
<i>Accuracy</i>			0.93	60
<i>Macro Avg</i>	0.94	0.93	0.93	60
<i>Weighted Avg</i>	0.94	0.93	0.93	60

Akurasi dihitung dengan menganalisis hasil prediksi dari model yang telah dilatih terhadap data *test*. Pada Tabel 4.3 bahwa total *test* data bernilai 60 dengan data yang benar diprediksi adalah 56, dan data yang salah diprediksi adalah 4, sehingga akurasinya adalah 93.333%.



1. Dilarang mengutip karya tulis ini, kecuali:
 - a. menyebutkan sumber sesuai kaidah kecendekian;
 - b. pengutipan hanya untuk keperluan pendidikan, penulisan karya ilmiah atau penelitian;
 - c. pengutipan tidak merugikan Universitas Pertamina.
2. Dilarang mempublikasikan atau memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa adanya izin dari Universitas Pertamina.

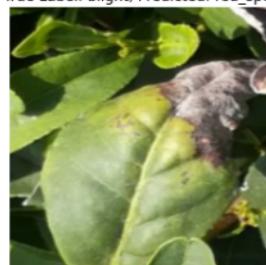
Tabel 4.3. Hasil Prediksi

<i>Total Test Data</i>	60
<i>Accurately Predicted Data</i>	56
<i>Wrongly Predicted Data</i>	4
<i>Accuracy</i>	93.333%

True Label: blight, Predicted: red_spot



True Label: blight, Predicted: red_spot



Gambar 4.6. Gambar yang Salah Diprediksi

True Label: red_spot, Predicted: red_scab



True Label: red_spot, Predicted: red_scab

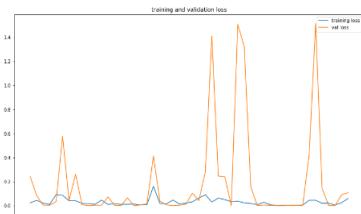
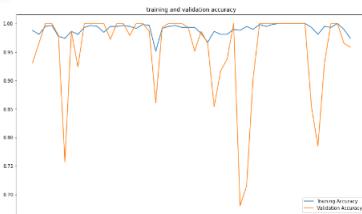


Gambar 4.7. Gambar yang Salah Diprediksi (2)

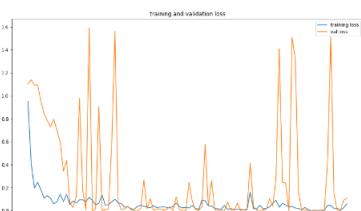
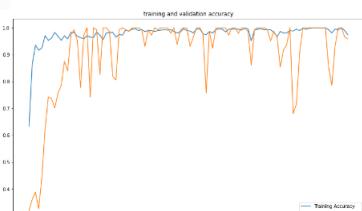
Langkah terakhir adalah menampilkan gambar yang salah diprediksi oleh model. Pada Gambar 4.6 dan 4.7 terlihat bahwa gambar pertama diprediksi oleh model sebagai *red spot*. Namun gambar

tersebut sebenarnya adalah teh yang berpenyakit *blight*. Lalu kedua gambar berikutnya diprediksi sebagai *red scab*, namun gambar tersebut sebenarnya daun teh yang berpenyakit *red spot*.

4.2.2 Percobaan Epoch = 100



Gambar 4.8. Evaluasi Score (50)



Gambar 4.9. Evaluasi Score (100)

Pada Gambar 4.8 terlihat nilai akurasi dan *loss* dari data *train* dan validasi sampai *epoch* 50 di mana nilai akurasi dan *loss* tidak stabil. Begitu juga pada Gambar 4.9 terlihat bahwa nilai akurasi dan *loss* dari data *train* dan validasi sampai *epoch* 100 di mana nilai akurasi dan *loss* tidak stabil juga.

Tabel 4.4. Evaluasi Score

<i>Test Accuracy</i>	0.800
<i>Test Loss</i>	1.182

Berbeda dengan tahapan proses sebelumnya, ketika percobaan pada pelatihan model dengan nilai *epoch* 100. Dapat dilihat pada gambar diatas bahwa nilai akurasi dan *loss* terhadap data *train* dan validasi tidak stabil, dapat dilihat bahwa fluktuasi tidak membaik dari awal sampai akhir nilai *epoch*. Dapat dilihat juga pada Tabel 4.1 nilai akurasi yaitu 0.80 dan *loss* yaitu 1.182.

1. Dilarang mengutip karya tulis ini, kecuali:
 - a. menyebutkan sumber sesuai kaidah akademika;
 - b. pengutipan hanya untuk keperluan penerjemahan, penulisan karya ilmiah, atau penelitian;
 - c. pengutipan tidak merugikan Universitas Pertamina.
2. Dilarang mempublikasikan atau memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa adanya izin dari Universitas Pertamina.

Tabel 4.5. Laporan Evaluasi *Epoch*=100

	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>	<i>Support</i>
0	1.00	0.65	0.79	20
1	0.80	1.00	0.89	20
2	0.68	0.75	0.71	20
<i>Accuracy</i>			0.80	60
<i>Macro Avg</i>	0.83	0.83	0.80	60
<i>Weighted Avg</i>	0.83	0.83	0.80	60

1.

- b. Informasi dari Tabel 4.5 rata - rata akurasinya adalah 0.80 (80%), lalu ada *f1-score* terhadap masing - masing kelas. Untuk kelas 0 (pertama:*blight*) adalah 0.79 (79%), kelas 1 (kedua:*red scab*) 0.89 (89%), dan kelas 2 (ketiga:*red spot*) 0.71 (71%). Dibandingkan dengan percobaan sebelumnya, pada *epoch* 50, terlihat jelas bahwa ini lebih buruk.

Tabel 4.6. Hasil Prediksi

<i>Total Test Data</i>	60
<i>Accurately Predicted Data</i>	48
<i>Wrongly Predicted Data</i>	12
<i>Accuracy</i>	80.00%

Saat proses memprediksi data *test* juga terlihat hasil bahwa dari total 60 data gambar yang diprediksi, hanya 48 gambar yang diprediksi dengan benar, 12 gambar salah diprediksi oleh model.

- a. dilakukan pengutip karya tulis ini, kecuali:
 - b. penyalinan hanya untuk keperluan pendidikan, penulisan karya ilmiah atau penelitian;
 - c. pengutipan tidak mengiklan Universitas Pertamina.
- 2. Dilarang mempublikasikan atau memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa adanya izin dari Universitas Pertamina.

1. Dilarang mengutip karya tulis ini, kecuali:
 - a. menyebutkan sumber sesuai kaidah kecendekian;
 - b. pengutipan hanya untuk keperluan pendidikan, penulisan karya ilmiah atau penelitian;
 - c. pengutipan tidak merugikan Universitas Pertamina.
2. Dilarang mempublikasikan atau memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa adanya izin dari Universitas Pertamina.



Gambar 4.10. Gambar yang Salah Diprediksi



Gambar 4.11. Gambar yang Salah Diprediksi (2)

1. Dilarang mengutip karya tulis ini, kecuali:
 - a. menyebutkan sumber sesuai kaidah kecendekian;
 - b. pengutipan hanya untuk keperluan pendidikan, penulisan karya ilmiah atau penelitian;
 - c. pengutipan tidak merugikan Universitas Pertamina.
2. Dilarang mempublikasikan atau memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa adanya izin dari Universitas Pertamina.

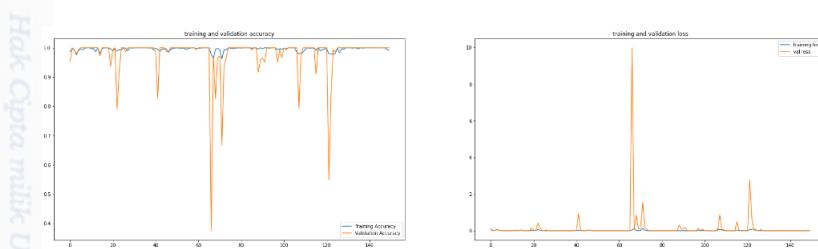


Gambar 4.12. Gambar yang Salah Diprediksi (3)

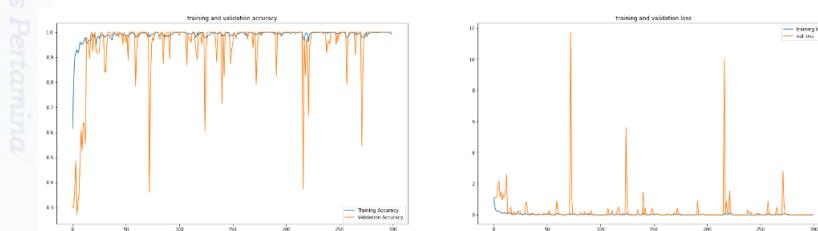


Gambar 4.13. Gambar yang Salah Diprediksi (4)

4.2.3 Percobaan Epoch = 300



Gambar 4.14. Evaluasi Score (150)



Gambar 4.15. Evaluasi Score (300)

Pada Gambar 4.14 terlihat nilai akurasi dan *loss* dari data *train* dan validasi sampai *epoch* 150 di mana nilai *loss* mulai membaik setelah *epoch* 60. Pada Gambar 4.15 terlihat bahwa nilai akurasi dan *loss* dari data *train* dan validasi sampai *epoch* 300 di mana nilai akurasi masih tidak cukup stabil namun pada nilai *loss* stabil pada saat sekitar *epoch* 125 sampai 210.

Tabel 4.7. Evaluasi Score

<i>Test Accuracy</i>	0.899
<i>Test Loss</i>	0.304

Berbeda juga dengan tahapan proses sebelumnya, percobaan pada pelatihan model dengan nilai *epoch* 300, memiliki nilai akurasi dan *loss* terhadap data *train* dan validasi tidak stabil, yang dapat dilihat bahwa fluktuasi tidak membaik dari awal sampai akhir nilai *epoch*. Dapat dilihat juga pada Tabel 4.1 nilai akurasi yaitu 0.899 dan *loss* yaitu 0.304.

Tabel 4.8. Laporan Evaluasi *Epoch*=300

	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>	<i>Support</i>
0	1.00	0.85	0.92	20
1	0.87	1.00	0.93	20
2	0.85	0.85	0.85	20
<i>Accuracy</i>			0.90	60
<i>Macro Avg</i>	0.91	0.90	0.90	60
<i>Weighted Avg</i>	0.91	0.90	0.90	60

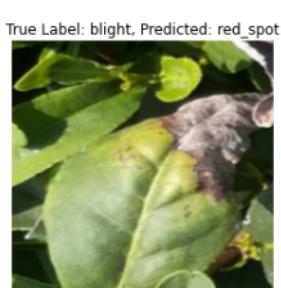
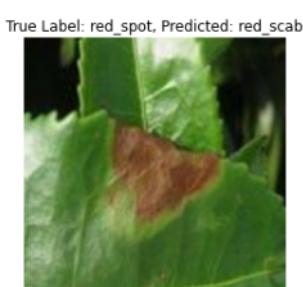
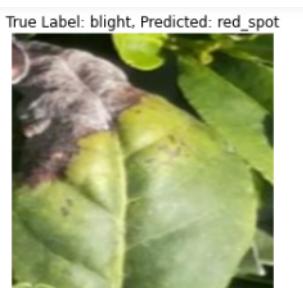
1. Dilarang mengutip karya tulis ini, kecuali:
 - a. menyebutkan sumber sifatnya; pendidikan, penulisan karya ilmiah atau penelitian;
 - b. pengutipan hanya untuk keperluan pendidikan, penulisan karya ilmiah atau penelitian;
 - c. pengutipan tidak mengikuti Universitas Pertamina.
2. Dilarang mempublikasikan atau memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa adanya izin dari Universitas Pertamina.

Informasi dari Tabel 4.8 rata - rata akurasinya adalah 0.90 (90%), lalu ada *f1-score* terhadap masing - masing kelas. Untuk kelas 0 (pertama:*blight*) adalah 0.92 (92%), kelas 1 (kedua:*red scab*) adalah 0.93 (93%), dan kelas 2 (ketiga:*red spot*) adalah 0.85 (85%). Dibandingkan dengan percobaan sebelumnya pada *epoch* 50, terlihat jelas bahwa ini lebih buruk, namun sedikit lebih baik dibandingkan dengan percobaan *epoch* 100.

Tabel 4.9. Hasil Prediksi *Epoch*=300

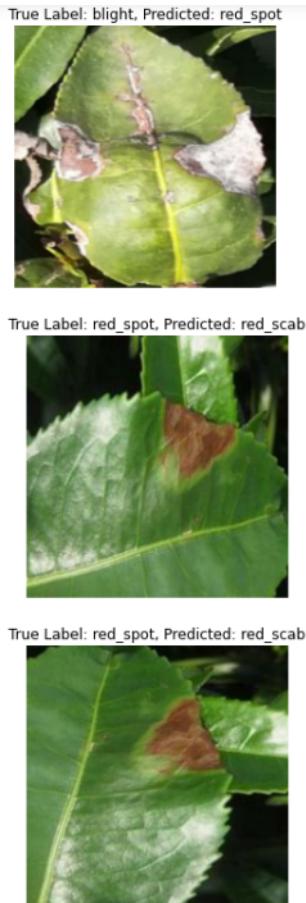
<i>Total Test Data</i>	60
<i>Accurately Predicted Data</i>	54
<i>Wrongly Predicted Data</i>	6
<i>Accuracy</i>	90.00%

Saat proses memprediksi data *test* juga terlihat hasil bahwa dari total 60 data gambar yang diprediksi, hanya 54 gambar yang diprediksi dengan benar, 6 gambar salah diprediksi oleh model.



Gambar 4.16. Gambar yang Salah Diprediksi

1. Dilarang mengutip karya tulis ini, kecuali:
 - a. menyebutkan sumber sesuai kaidah kecepatan;
 - b. pengutipan hanya untuk keperluan pendidikan, penulisan karya ilmiah atau penelitian;
 - c. pengutipan tidak merugikan Universitas Pertamina.
2. Dilarang mempublikasikan atau memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa adanya izin dari Universitas Pertamina.



Gambar 4.17. Gambar yang Salah Diprediksi (2)

4.2.4 Ringkasan

Tabel 4.10. Ringkasan per Epoch

Epoch	Test Accuracy	Test Loss	Wrongly Predicted Data
50	0.933	0.351	4
100	0.800	1.182	12
300	0.899	0.304	6

Dari percobaan - percobaan diatas dapat disimpulkan bahwa jumlah *epoch* tertentu dapat memberikan solusi mendekati optimal. Peningkatan ataupun penurunan jumlah *epoch* tidak selalu menghasilkan peningkatan kinerja model. Hasil terbaik diperoleh ketika percobaan dengan nilai *epoch* 50 dengan tingkat akurasi 93% dan hanya 4 gambar yang salah diprediksi.



5.1 Kesimpulan

Setelah dilakukan proses klasifikasi penyakit daun teh tersebut menggunakan DenseNet-121, dapat diketahui bahwa ada beberapa cara pencegahan agar tidak *overfitting* pada saat melakukan pelatihan model seperti augmentasi data, *dropout*, dan *callback (Model Checkpoint, ReduceLROnPlateau)*. Pada proses pelatihan juga dapat dilihat bahwa ketika nilai *epoch*-nya pada jumlah tertentu dapat memberikan solusi mendekati optimal. Namun, peningkatan ataupun penurunan nilai *epoch* tidak selalu menghasilkan peningkatan kinerja model, dan peningkatan nilai *epoch* juga berpotensi menyebabkan *overfitting* jika tidak diimbangi dengan teknik-teknik pencegahan *overfitting*. Pelatihan model ini menggunakan nilai *epoch* 50.

Pada grafik evaluasi model, dapat dilihat bahwa nilai akurasi antara data *train* dan validasi seimbang saat *epoch* 20 sampai dengan 30. Ketika *epoch* 0 sampai dengan 20, akurasi data *train* dan validasi masih belum stabil atau dapat dikatakan model masih dalam tahap pembelajaran. Namun, fluktuasi atau keadaan yang belum stabil ini, secara perlahan mulai membaik sampai jumlah *epoch* seterusnya, yang artinya model dapat belajar dengan cukup baik. Sama hal nya terkait nilai *loss*, nilai *loss* dari kedua data tersebut mulai stabil ketika berada di *epoch* 20 sampai dengan 30. Seterusnya fluktuasinya mulai membaik, hanya saja sebelum *epoch* 40 terjadi peningkatan nilai *val loss* yang signifikan.

Ketika model melakukan prediksi terhadap data *test*, didapatkan informasi pada laporan klasifikasi bahwa nilai *f1-score* terhadap masing - masing kelas yaitu untuk kelas 0 (pertama:*blight*) di angka 0.95 (95%), kelas 1 (kedua:*red scab*) di angka 0.95 (95%), kelas 2 (ketiga:*red spot*) di angka 0.90 (90%), dan *accuracy* di angka 0.93 (93%). Dapat disimpulkan bahwa model dapat memprediksi data *test* dengan sangat baik yaitu dari pembahasan di atas dijelaskan bahwa model hanya salah memprediksi 4 gambar.

5.2 Saran

Adapun beberapa saran terkait penelitian ini agar penelitian dapat berkembang jauh lebih baik, di antaranya:

1. Dataset dikumpulkan oleh pihak peneliti itu sendiri, contohnya foto langsung tanaman teh di kebun teh. Dengan mengumpulkan data sendiri, dapat dipastikan bahwa data yang digunakan cocok dengan karakteristik daun teh yang ingin diklasifikasikan.
2. Melakukan eksplorasi metode - metode terhadap klasifikasi daun teh. Dengan mengeksplorasi metode - metode tersebut, dapat dilakukan perbandingan dan dapat diambil kesimpulan metode apa yang lebih cocok untuk klasifikasi tanaman itu sendiri.



DAFTAR PUSTAKA

- Annamalai, M. (2018). Understanding the Classification Report in sklearn. <https://muthu.co/understanding-the-classification-report-in-sklearn/>. [Accessed on August 29, 2023].
- Bao, W., Fan, T., Hu, G., Liang, D., and Li, H. (2022). Detection and identification of tea leaf diseases based on ax-retinanet. *Scientific Reports*, 12(1):2183.
- Brown, S. (2021). Machine Learning Explained. <https://mitsloan.mit.edu/ideas-made-to-matter/machine-learning-explained>. [Accessed on June 7, 2023].
- Chouinard, J. (2022). Classification Report in scikit-learn. <https://www.jcchouinard.com/classification-report-in-scikit-learn/>. [Accessed on August 29, 2023].
- Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press. <http://www.deeplearningbook.org>.
- Hasan, N., Bao, Y., Shawon, A., and Huang, Y. (2021). Densenet convolutional neural networks application for predicting covid-19 using ct image. *SN computer science*, 2(5):389.
- Hu, G., Wu, H., Zhang, Y., and Wan, M. (2019). Data for: A low shot learning method for tea leaf's disease identification.
- Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. (2017). Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708.
- Kesehatan, K. (2022). Manfaat Teh bagi Tubuh. https://yankes.kemkes.go.id/view_artikel/1665/manfaat-teh-bagi-tubuh. [Accessed on June 8, 2023].
- Masoud, A. (2022). Understanding Micro, Macro, and Weighted Averages for scikit-learn Metrics in Multi-Class Classification with Example. <http://iamirmasoud.com/2022/06/19/understanding-micro-macro-and-weighted-averages-for-scikit-learn-metrics-in>. [Accessed on August 29, 2023].
- Meng, S., Wang, S., Zhou, T., and Shen, J. (2020). Identification of tea red leaf spot and tea red scab based on hybrid feature optimization. In *Journal of Physics: Conference Series*, volume 1486, page 052023. IOP Publishing.
- Saleem, M. H., Potgieter, J., and Arif, K. M. (2020). Plant disease classification: A comparative evaluation of convolutional neural networks and deep learning optimizers. *Plants*, 9(10):1319.
- Swaminathan, A., Varun, C., Kalaivani, S., et al. (2021). Multiple plant leaf disease classification using densenet-121 architecture. *International Journal of Electrical Engineering and Technology (IJEET)*, 12(5):38–57.
- Yamashita, R., Nishio, M., Do, R. K. G., and Togashi, K. (2018). Convolutional neural networks: an overview and application in radiology. *Insights into imaging*, 9:611–629.
- Zhong, Z., Zheng, M., Mai, H., Zhao, J., and Liu, X. (2020). Cancer image classification based on densenet model. In *Journal of Physics: Conference Series*, volume 1651, page 012143. IOP Publishing.





Form TA-4 Pendaftaran Sidang Tugas Akhir
FAKULTAS SAINS DAN ILMU KOMPUTER
PROGRAM STUDI ILMU KOMPUTER

Nama : Haikal Efendi
NIM : 105219027
Judul Tugas Akhir : Klasifikasi Penyakit Daun Teh Menggunakan DenseNet-121

Tanggal Seminar Kemajuan TA : 23 Juni 2023

Dilengkapi dengan karya tulis ini, kecuali:

- a. Tiga salinan draft Laporan Tugas Akhir (empat Salinan bila Pembimbing 2 orang)
- b. Bukti bahwa Laporan Tugas Akhir telah bebas plagiat
- c. Form Bimbingan Tugas Akhir (dengan jumlah bimbingan 8 kali atau lebih)
- d. Form bebas pinjam dari laboratorium

Jakarta, 10 Agustus 2023

Pendaftar,

Nama : Haikal Efendi

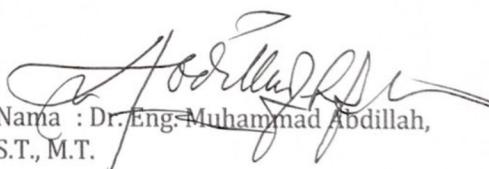
NIM : 105219027

Menyetujui,

Dosen Pembimbing 1,

Dosen Pembimbing 2,

Nama : Dr. Tasmi, S.Si., M.Si
NIP : 116109


Nama : Dr. Eng. Muhammad Abdillah,
S.T., M.T.

NIP : 116153

Dilarang mempublikasikan atau memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa adanya izin dari Universitas Pertamina.



Form TA-2 Bimbingan Tugas Akhir

FAKULTAS SAINS DAN ILMU KOMPUTER
PROGRAM STUDI ILMU KOMPUTER

No.	Hari/Tanggal: Kamis / 13 Juli 2023
-----	------------------------------------

Hal yang menjadi perhatian:

- Membahas mengenai kodingan terkait pelatihan model yang overfit

Ardy
Paraf Pembimbing:

Hari/Tanggal: Kamis / 27 Juli 2023

Hal yang menjadi perhatian:

- Membahas terkait kodingan bagaimana hasil dari prediksi data test.
- Membahas laporan yang harus dikerjakan

Ardy
Paraf Pembimbing:

2. Dilarang mengutip karya tulis ini, kecuali:
a. menyebutkan sumber sesuai kaidah kredibilitas;
b. pengutipan hanya untuk keperluan pendidikan, penulisan karya ilmiah atau penelitian;
c. pengutipan tidak merugikan Universitas Pertamina.



Form TA-2 Bimbingan Tugas Akhir

FAKULTAS SAINS DAN ILMU KOMPUTER
PROGRAM STUDI ILMU KOMPUTER

Nama Mahasiswa : Haikal Efendi

NIM: 105219027

Nama Pembimbing : Dr. Eng. Muhammad Abdillah, S.T., M.T.

NIP : 116153

2.

No.	Hari/Tanggal: Rabu / 7 Juni 2023
Hal yang menjadi perhatian:	
	<ul style="list-style-type: none">• Membahas laporan yang telah direvisi dari BAB 1 sampai dengan BAB 3.• Membahas mengapa memakai arsitektur DenseNet-121
Dilengungi Undang-Undang Copyright of Universitas Pertamina	
No.	Hari/Tanggal: Kamis / 15 Juni 2023
Hal yang menjadi perhatian:	
	<ul style="list-style-type: none">• Membahas bagaimana persiapan untuk seminar kemajuan• Membahas jadwal kosong untuk seminar kemajuan
Dilengungi Undang-Undang Copyright of Universitas Pertamina	
	 Paraf Pembimbing:
	 Paraf Pembimbing:

- b. pengutipan hanya untuk keperluan pendidikan, penulisan karya ilmiah atau penelitian;
- c. pengutipan tidak merugikan Universitas Pertamina.
- Dilarang mempublikasikan atau memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa adanya izin dari Universitas Pertamina.



Form TA-2 Bimbingan Tugas Akhir
FAKULTAS SAINS DAN ILMU KOMPUTER
PROGRAM STUDI ILMU KOMPUTER

No.	Hari/Tanggal: Kamis / 13 Juli 2023
Hal yang menjadi perhatian: 1. Membahas mengenai kodingan terkait pelatihan model yang overfit	
Paraf Pembimbing:	
No.	Hari/Tanggal: Kamis / 27 Juli 2023
Hal yang menjadi perhatian: 1. Membahas terkait kodingan bagaimana hasil dari prediksi data test. 2. Membahas laporan yang harus dikerjakan a. menyebutkan sumber sesuai kaidah mendeklaikan; b. pengutipan hanya untuk keperluan pendidikan, penulisan karya ilmiah atau penelitian; c. pengutipan tidak merugikan Universitas Pertamina. 2. Dilarang mempublikasikan atau memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa adanya izin dari Universitas Pertamina.	
Paraf Pembimbing:	

 <i>Dilindungi Undang-Undang</i> © Copyright of Universitas Pertamina	<p>Form TA-2 Bimbingan Tugas Akhir</p> <p>FAKULTAS SAINS DAN ILMU KOMPUTER</p> <p>PROGRAM STUDI ILMU KOMPUTER</p>
No.	Hari/Tanggal: Kamis / 5 Juni 2023
Hal yang menjadi perhatian:	
<ul style="list-style-type: none">• Membahas mengenai kodingan model awal memakai arsitektur DenseNet.• Melakukan revisi Latar Belakang dan Tinjauan Pustaka..	
Paraf Pembimbing: 	
No.	Hari/Tanggal: Rabu / 7 Juni 2023
Hal yang menjadi perhatian:	
<ul style="list-style-type: none">• Membahas laporan yang telah direvisi dari BAB 1 sampai dengan BAB 3.• Membahas mengapa memakai arsitektur DenseNet-121	
Paraf Pembimbing: 	

2. Dilarang mengutip karya tulis ini, kecuali:
a. menyebutkan sumber sesuai kaidao.
b. pengutipan hanya untuk keperluan pendidikan, penulisan karya ilmiah atau penelitian;
c. pengutipan tidak merugikan Universitas Pertamina.
Dilarang mempublikasikan atau memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa adanya izin dari Universitas Pertamina.



Form TA-2 Bimbingan Tugas Akhir
FAKULTAS SAINS DAN ILMU KOMPUTER
PROGRAM STUDI ILMU KOMPUTER

No.	Hari/Tanggal: Senin / 14 April 2023
-----	-------------------------------------

Hal yang menjadi perhatian:

- Masih membahas terkait dataset karena belum menemukan yang cocok.
- Serta membahas paper/artikel untuk tinjauan pustaka.

Membahas apa saja yang akan dikerjakan pada BAB 1: Pendahuluan.

Paraf Pembimbing:

	Hari/Tanggal: Selasa / 16 Mei 2023
--	------------------------------------

Hal yang menjadi perhatian:

- Membahas mengenai kodingan model awal
- Melakukan revisi latar belakang.
- Mencari informasi mengenai artikel yang sesuai dengan topik TA.

Paraf Pembimbing:

2. Dilarang mengutip karya tulis ini, kecuali:
- a. menyebutkan sumber sesuai kaidah Hacendekiaan;
 - b. pengutipan hanya untuk keperluan pendidikan, penulisan karya ilmiah atau penelitian;
 - c. pengutipan tidak merugikan Universitas Pertamina.
- Dilarang mempublikasikan atau memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa adanya izin dari Universitas Pertamina.



Form TA-2 Bimbingan Tugas Akhir
FAKULTAS SAINS DAN ILMU KOMPUTER
PROGRAM STUDI ILMU KOMPUTER

Nama Mahasiswa : Haikal Efendi

NIM : 105219027

Nama Pembimbing : Dr. Tasmi, S.Si, M.Si

NIP : 116109

No.	Hari/Tanggal: Kamis / 29 Desember 2022
Hal yang menjadi perhatian: Dilarang mengutip karya tulis ini, kecuali: Hal yang menjadi perhatian:	
No.	Hari/Tanggal: Selasa / 14 Maret 2023
Hal yang menjadi perhatian: Dilarang mengutip karya tulis ini, kecuali: Hal yang menjadi perhatian: <ul style="list-style-type: none">• Membahas artikel – artikel yang sesuai dengan topik.• Membahas mengapa mengambil topik Klasifikasi penyakit tanaman daun teh.• Membahas metode untuk klasifikasi.	

Paraf Pembimbing:

Paraf Pembimbing:

2. Dilarang mempublikasikan atau memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa adanya izin dari Universitas Pertamina.
- b. pengutipan hanya untuk keperluan pendidikan, penulisan karya ilmiah atau penelitian;
- c. pengutipan tidak merugikan Universitas Pertamina.