

**Implementasi Kecerdasan Buatan Untuk Kalibrasi Citra  
Tanaman pada Proses Prediksi Kandungan Pigmen  
Fontosintesis Secara Non Destruktif dan In-Situ**



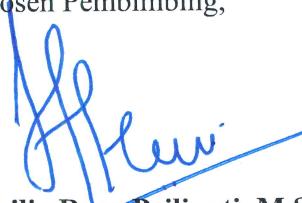
**MARCELINO CENTAURI DWI PRASETYO  
NIM : 311810020**

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS MA CHUNG  
MALANG  
2021**

LEMBAR PENGESAHAN  
PRAKTIK KERJA LAPANGAN

IMPLEMENTASI KECERDASAN BUATAN UNTUK KALIBRASI CITRA  
TANAMAN PADA PROSES PREDIKSI KANDUNGAN PIGMEN  
FOTOSINTESIS SECARA NON DESTRUKTIF DAN IN-SITU

Oleh:  
**MARCELINO CENTAURI DWI PRASETYO**  
**NIM. 311810020**

dari:  
Dosen Pembimbing,  
  
Dr. Kestrilia Rega Prilianti, M.Si  
NIP. 20120035

Dekan Fakultas Sains dan Teknologi  
  
Dr. Kestrilia Rega Prilianti, M.Si  
NIP. 201200

## **KATA PENGANTAR**

Puji syukur dipanjangkan kepada Tuhan Yang Maha Esa atas restu-Nya sehingga dapat menyelesaikan tugas dan laporan Praktik Kerja Lapangan dengan baik sampai akhir.

Laporan Praktik Kerja Lapangan yang berjudul “Implementasi Kecerdasan Buatan Untuk Kalibrasi Citra Tanaman pada Proses Prediksi Kandungan Pigmen Fontosintesis Secara Non Destruktif dan In-Situ” dibuat untuk memenuhi mata kuliah wajib sebagai salah satu mahasiswa dari Universitas Ma Chung Malang. Pembuatan laporan ini juga dilakukan sebagai rekapan mengenai hal-hal yang telah dikerjakan selama kegiatan Praktik Kerja Lapangan berlangsung di satu semester.

Pada kesempatan ini penulis ingin mengucapkan terima kasih kepada semua pihak yang sudah membantu penulis dalam proses pembuatan praktik kerja lapangan ini. Ucapan terima kasih juga saya ucapkan kepada:

1. Ibu Dr. Kestrilia Rega Prilianti, M.Si., Selaku Dekan Fakultas Sains dan Teknologi serta dosen pembimbing dalam penggerjaan proyek praktik kerja lapangan,
2. Bapak Hendry Setiawan, ST., M. Kom., Selaku Kepala Program Studi Teknik Informatika Universitas Ma Chung,
3. Orang tua terkasih yang selalu mendukung penulis dan memberikan semangat untuk menyelesaikan praktik kerja lapangan ini,
4. Dan teman-teman yang sudah saling membantu dalam penggerjaan praktik kerja lapangan ini.

Hingga selesai, proyek Praktik Kerja Lapangan ini tetap dapat dilakukan secara lancar meskipun proses penggerjaannya dilakukan secara daring.

Malang, 30 Desember 2021

Marcelino Centauri

311810020

## DAFTAR ISI

LEMBAR PENGESAHAN .....	i
KATA PENGANTAR .....	ii
DAFTAR ISI.....	iii
DAFTAR PERSAMAAN .....	v
DAFTAR GAMBAR .....	vi
DAFTAR TABEL.....	vii
BAB I .....	1
PENDAHULUAN .....	1
1.1 Latar Belakang Masalah .....	1
1.2 Identifikasi Masalah .....	3
1.3 Batasan Masalah.....	3
1.4 Rumusan Masalah .....	3
1.5 Tujuan.....	4
1.6 Manfaat.....	4
BAB II.....	5
GAMBARAN UMUM PERUSAHAAN.....	5
2.1 Program Studi Teknik Informatika Universitas Ma Chung.....	5
2.2 Roadmap Penelitian Bidang <i>Precision Agriculture</i> .....	8
BAB III .....	12
TINJAUAN PUSTAKA .....	12
3.1 Pigmen Tanaman .....	12
3.2 Color Correction .....	14
3.3 Algoritma K-nearest <i>neighbors</i> .....	14
3.4 <i>Deep neural network</i> .....	15

3.5 White-Balance .....	18
3.6 Principal Component Analysis (PCA).....	19
3.7 Arsitektur Multi- <i>Decoder</i> .....	20
3.8 Pengolahan Citra Digital .....	21
3.9 Python .....	21
3.10 Color Transformation .....	22
3.11 Github Algoritma KNN dan DNN.....	22
3.12 Ringkasan Jurnal Tentang <i>When Color Constancy Goes Wrong</i> .....	23
3.13 Ringkasan Jurnal Tentang <i>Deep White-Balance</i> .....	26
<b>BAB IV .....</b>	<b>28</b>
<b>HASIL DAN PEMBAHASAN.....</b>	<b>28</b>
4.1 Proses Penggerjaan .....	28
4.2 Studi Literatur.....	29
4.3 Pengumpulan Dataset .....	32
4.4 Proses <i>Training</i> dan <i>Testing</i> Model.....	39
4.5 Proses Pengujian.....	40
4.6 Pengukuran Hasil.....	48
<b>BAB V.....</b>	<b>60</b>
<b>KESIMPULAN DAN SARAN.....</b>	<b>60</b>
5.1 Kesimpulan .....	60
5.2 Saran .....	60
<b>DAFTAR PUSTAKA .....</b>	<b>62</b>
<b>LAMPIRAN .....</b>	<b>65</b>

## DAFTAR GAMBAR

Gambar 2.1 Peta Jalan Penelitian <i>Precision Agriculture</i> di UMC.....	9
Gambar 3.1 Grafik Absorbansi Pigmen Tanaman .....	14
Gambar 3.2 Algoritma K-nearest neighbor.....	15
Gambar 3.3 Posisi <i>Deep neural network</i> pada dunia AI.....	16
Gambar 3.4 Neuron pada <i>Artificial Neural Network</i> .....	16
Gambar 3.5 Arsitektur <i>Deep neural networks</i> .....	18
Gambar 3.6 <i>White Balance</i> .....	19
Gambar 3.7 Ilustrasi PCA .....	20
Gambar 3.8 Perbedaan Penggunaan <i>White-balance</i> .....	23
Gambar 3.9 Diagram Prosedur Algoritma KNN.....	26
Gambar 3.10 Perbedaan Penggunaan Algoritma KNN dan DNN .....	27
Gambar 4.1 Tahapan Pengerjaan .....	28
Gambar 4.2 Arsitektur Model DNN .....	32
Gambar 4.3 Augmentasi pada Temperatur Warna.....	39

## DAFTAR TABEL

Tabel 2.1 <i>Roadmap</i> Penelitian Kelompok Riset Precision Agriculture .....	10
Tabel 4.1 Contoh Dataset yang Digunakan pada KNN dan DNN .....	34
Tabel 4.2 Tingkat Pencahayaan Daun <i>Duranta Erecta</i> .....	35
Tabel 4.3 Contoh Dataset <i>Duranta Erecta</i> (DE 1).....	36
Tabel 4.3 (Lanjutan).....	37
Tabel 4.4 Tingkat Pencahayaan Daun <i>Piper Betle</i> .....	37
Tabel 4.5 Contoh Dataset <i>Piper Betle</i> .....	38
Tabel 4.6 Hasil Kalibrasi Algoritma KNN <i>Duranta Erecta</i> (DE 1) .....	41
Tabel 4.6 (Lanjutan).....	42
Tabel 4.7 Hasil Kalibrasi Algoritma KNN <i>Piper Betle</i> (PB 1).....	43
Tabel 4.8 Hasil Kalibrasi Algoritma DNN <i>Duranta Erecta</i> (DE 1) .....	45
Tabel 4.9 Hasil Kalibrasi Algoritma DNN <i>Piper Betle</i> (PB 1).....	46
Tabel 4.10 Hasil <i>Similiarity Original Duranta Erecta</i> .....	48
Tabel 4.10 (Lanjutan).....	49
Tabel 4.11 Hasil <i>Similiarity Original Piper Betle</i> .....	49
Tabel 4.12 Hasil <i>Similiarity</i> KNN Model Set 1 <i>Duranta Erecta</i> .....	50
Tabel 4.13 Hasil <i>Similiarity</i> KNN Model Set 2 <i>Duranta Erecta</i> .....	51
Tabel 4.14 Hasil <i>Similiarity</i> KNN Model Set 1 <i>Piper Betle</i> .....	52
Tabel 4.15 Hasil <i>Similiarity</i> KNN Model Set 2 <i>Piper Betle</i> .....	52
Tabel 4.15 (Lanjutan).....	53
Tabel 4.16 Hasil <i>Similiarity</i> DNN <i>Duranta Erecta</i> .....	53
Tabel 4.17 Hasil <i>Similiarity</i> DNN <i>Piper Betle</i> .....	54
Tabel 4.18 Hasil Prediksi Pigmen <i>Duranta Erecta</i> Kalibrasi KNN Set 1 .....	56
Tabel 4.19 Hasil Prediksi Pigmen <i>Duranta Erecta</i> Kalibrasi KNN Set 2 .....	56

Tabel 4.20 Hasil Prediksi Pigmen <i>Duranta Erecta</i> Kalibrasi DNN .....	57
Tabel 4.21 Hasil Prediksi Pigmen <i>Piper Betle</i> Kalibrasi KNN Set 1 .....	57
Tabel 4.22 Hasil Prediksi Pigmen <i>Piper Betle</i> Kalibrasi KNN Set 2 .....	58
Tabel 4.23 Hasil Prediksi Pigmen <i>Piper Betle</i> Kalibrasi DNN .....	58

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang Masalah**

Aplikasi *leaf piction* adalah aplikasi pengenalan warna pigmen pada daun suatu tumbuhan berbasis android. Aplikasi ini dikembangkan dengan tujuan untuk membantu pengguna dalam mengidentifikasi jumlah pigmen yang terdapat pada tanaman sehingga dapat ditentukan sehat tidaknya suatu tanaman tersebut. Sebelumnya aplikasi ini telah dikembangkan oleh kelompok riset *Precision Agriculture* (PA) dari Universitas Ma Chung dengan ketua pembimbingnya adalah Dr. Kestrilia Rega Prilianti, M.Si. Aplikasi ini akan mengidentifikasi warna pigmen yang terbagi menjadi 3 yaitu *klorofil*, *carotenoid* dan *antocyanin*. Pengenalan pigmen ini dikembangkan dengan menggunakan kecerdasan buatan untuk memprediksi warna-warna pigmen yang dimiliki suatu tumbuhan. Melalui penggunaan kecerdasan buatan, aplikasi ini mampu melakukan identifikasi tanpa harus merusak tanaman (non destruktif) sehingga warna daun akan lebih terjaga. Penggunaan aplikasi berjalan lancar ketika diuji coba di sebuah lab dengan kondisi cahaya tertentu dan menunjukkan akurasi yang baik namun setelah dilakukan pengujian di lapangan terdapat penurunan akurasi dikarenakan aplikasi belum invariant terhadap perbedaan cahaya dan spek kamera yang digunakan untuk mengambil gambar.

Identifikasi pigmen tanaman adalah proses untuk mencari jumlah pigmen yang ada di dalam sebuah tumbuhan, tujuannya untuk mengetahui kesehatan suatu tumbuhan tersebut. Identifikasi ini juga akan membantu para petani dalam menentukan kondisi tanamannya apakah sakit atau tidak. Dalam melakukan identifikasi, sebuah citra daun pada tumbuhan akan digunakan dan diperiksa dengan bantuan *deep learning* untuk mendapatkan jumlah masing-masing pigmen pada daun tersebut. Ketika melakukan pengambilan gambar, kondisi lingkungan tempat pengambilan juga perlu diperhatikan karena akan mempengaruhi kuantifikasi kandungan pigmen citra. Hal tersebut menjadi permasalahan karena untuk mendapatkan suatu pencahayaan yang sesuai, pengambilan gambar perlu dilakukan

di kondisi cahaya yang benar-benar sama tidak boleh terdapat perbedaan jumlah cahaya yang masuk. Oleh karena itu diperlukanlah suatu metode kalibrasi citra untuk menyamakan pencahayaan yang tertangkap ketika melakukan pengambilan citra. Kalibrasi ini diharapkan dapat membantu pengambilan gambar tanpa harus dilakukan di dalam ruangan dengan kondisi cahaya tertentu, melainkan dapat dilakukan dimana saja atau di tempat pengambilan dilakukan (*in situ*) tanpa harus mempedulikan kondisi cahaya di lingkungan tersebut.

Pengolahan kalibrasi kamera memiliki peranan yang sangat penting dalam proses pengambilan suatu citra gambar. Suatu kamera memerlukan kalibrasi gambar agar warna yang didapatkan dapat sesuai dengan mata manusia. Kalibrasi kamera juga berguna dalam melakukan normalisasi warna pada suatu citra gambar. Kalibrasi kamera juga berperan dalam mengubah kontras warna yang ada pada suatu gambar agar gambar yang dimiliki memiliki kontras warna sesuai untuk dilihat mata manusia (Bevans & Chattman, 2004). Hal tersebut biasanya digunakan untuk mengatasi permasalahan pada pengambilan gambar di malam hari dan ketika di dalam air. Gambar yang tidak dilakukan kalibrasi akan terlihat gelap dan terkadang tidak jelas ketika dilihat oleh mata manusia karena kurangnya sumber cahaya yang masuk maupun berlebihan (Bianco & Muzzupappa, 2015). Pengambilan gambar tanpa menggunakan kalibrasi kamera akan menghasilkan kontras warna yang berbeda-beda contohnya ketika difoto pada sore hari dimana kontras cahayanya akan terlihat sedikit jingga dengan gambar yang difoto pada pagi hari dimana kontrasnya akan lebih mengarah kebiru-biruan (Gasparini & Schettini, 2014). Hal tersebut juga akan mempengaruhi kuantifikasi kandungan pigmen daun sehingga prediksi pigmen yang ditampilkan akan menunjukkan hasil yang berbeda-beda untuk setiap kondisi cahaya yang juga berbeda. Oleh karena itu citra daun tersebut perlu dilakukan kalibrasi agar mendapatkan hasil yang sesuai meskipun ditangkap pada kondisi cahaya yang berbeda. Pada kasus ini, kalibrasi kamera akan digunakan untuk menyamakan cahaya lingkungan yang terdapat pada suatu citra daun tumbuhan. Metode kalibrasi juga memerlukan bantuan dari kecerdasan buatan untuk mengidentifikasi pola suatu cahaya yang masuk pada citra, dimana nantinya akan dikalibrasi untuk disamakan pencahayaannya. Salah satu metode kalibrasinya adalah *white-balancing*.

## 1.2 Identifikasi Masalah

Berdasarkan latar belakang yang telah dipaparkan dapat dilakukan identifikasi masalah bahwa aplikasi *leaf piction* masih belum invarian terhadap cahaya dan spek kamera yang digunakan di lapangan sehingga terjadi penurunan akurasi ketika diuji coba. Akurasi yang dimaksud ini adalah tingkat keberhasilan algoritma dari aplikasi *leaf piction* dalam memprediksi kandungan pigmen yang terdapat pada suatu citra daun tanaman dengan cara membandingkan nilai asli kandungan pigmen yang diuji secara lab. Semakin bagus akurasinya maka prediksi yang dilakukan oleh aplikasi semakin tepat dengan hasil uji labnya.

## 1.3 Batasan Masalah

Adapun batasan masalah dalam penggerjaan proyek ini adalah sebagai berikut.

- Teknik kalibrasi yang dilakukan hanya akan terfokus pada citra daun tanaman saja.
- Teknik kalibrasi yang dilakukan tidak menggunakan segmentasi gambar untuk daun.
- Proyek yang dikerjakan hanya akan melakukan proses pengujian algoritma kalibrasi saja, tidak sampai diimplementasikan kedalam aplikasi *leaf piction*.
- Teknik kalibrasi yang digunakan mengambil dari referensi peneliti di internet dan tidak dibuat sendiri mulai dari awal.

## 1.4 Rumusan Masalah

Berdasarkan identifikasi masalah yang telah dibahas, maka rumusan masalah dari proyek ini adalah sebagai berikut.

- Bagaimana cara mengembangkan metode kalibrasi “unsupervised” yang tepat digunakan untuk citra gambar daun tanaman?
- Seberapa baik teknik kalibrasi yang digunakan pada citra tanaman daun?

## 1.5 Tujuan

Tujuan dari penggerjaan proyek ini adalah sebagai berikut.

- Melakukan kalibrasi tanaman daun tanpa harus melakukan segmentasi atau pengolahan *background* gambar dengan salah satu metodenya adalah *white-balancing*.
- Mengembangkan teknik kalibrasi menggunakan algoritma KNN dan *Deep Learning* dalam melakukan kalibrasi sehingga mendapatkan hasil prediksi yang sesuai.

## 1.6 Manfaat

Manfaat dari penggerjaan proyek ini dibagi menjadi beberapa bagian sebagai berikut.

- Bagi masyarakat, dapat menggunakan aplikasi *leaf piction* tanpa harus mempedulikan kondisi cahaya yang terjadi di lapangan.
- Bagi Universitas khususnya Program Studi Teknik Informatika, dapat memberikan pengalaman bagi mahasiswa untuk siap di dunia kerja khususnya dalam bentuk praktik kerja lapangan.
- Bagi penulis, dapat menambah wawasan tentang kalibrasi warna dengan cara klasifikasi dan pentingnya pengaruh suatu cahaya dalam bidang pengambilan gambar atau perfotoan.

## **BAB II**

### **GAMBARAN UMUM PERUSAHAAN**

#### **2.1 Program Studi Teknik Informatika Universitas Ma Chung**

Universitas Ma Chung adalah perguruan tinggi swasta Indonesia yang ada di bawah naungan Yayasan Harapan Bangsa Sejahtera. Universitas ini sudah mulai berdiri dan diresmikan sejak 7 Juli 2007. Pada saat ini gedung utama kampus tersebut terletak di Villa Puncak Tidar N-01, Karangwidoro, Dau, Malang, Jawa, Indonesia.

Visi universitas Ma Chung adalah memuliakan Tuhan melalui akhlak, pengetahuan, dan kontribusi nyata sebagai insan akademik yang berdaya cipta. Kemudian misi Universitas Ma Chung adalah sebagai berikut.

- a. Menyelenggarakan Tri Dharma Perguruan Tinggi yaitu pendidikan dan pengajaran tinggi, penelitian, dan pengabdian kepada masyarakat secara berkualitas, fokus, dan sesuai dengan kebutuhan masyarakat kini dan akan datang.
- b. Membentuk dan mengembangkan angkatan-angkatan motivator dan pemimpin masyarakat yang memiliki potensi dan kapasitas moral yang luhur, berjiwa kepemimpinan dan kewirausahaan yang betitik berat pada pembentukan akhlak dan kepribadian unggul, rendah hati, melayani, dan berkontribusi sebagai manusia yang utuh.
- c. Mendorong dan mengembangkan sikap serta pemikiran yang kritis-prinsipil dan kreatif-realistik berdasarkan kepekaan hati nurani yang luhur.
- d. Menghasilkan lulusan siap pakai yang berkualitas tinggi yang mampu bersaing di pasar global.
- e. Berperan aktif dalam meningkatkan peradaban dunia dengan menghasilkan lulusan yang berwawasan global, toleran, dan cinta damai, serta produktif dalam menghasilkan karya cipta yang mendukung peningkatan martabat manusia global

- f. Melaksanakan pengelolaan perguruan tinggi berdasarkan prinsip ekonomis dan akuntabilitas.

Universitas Ma Chung juga memiliki pondasi yang dinamakan dengan 12 nilai Ma Chung. Isi dari 12 nilai Ma Chung tersebut adalah sebagai berikut.

1) Orisinal

Universitas Ma Chung percaya bahwa setiap orang adalah istimewa sehingga setiap orang harus mampu menjadi dirinya sendiri, dan tidak mengekor orang lain. Oleh karena itu, segenap civitas akademika Universitas Ma Chung harus mempunyai komitmen untuk berani tampil beda dan mengedepankan orisinalitas dalam karya dan karsa.

2) Terpercaya

Universitas Ma Chung menjunjung tinggi kejujuran dalam berpikir, bertindak, dan berbicara dalam upaya membangun institusi dan civitas akademika yang memiliki kredibilitas tinggi, terhormat, dapat diandalkan, dan terpercaya.

3) Gigih

Universitas Ma Chung percaya bahwa kesabaran dan kegigihan mampu mengalahkan semua tantangan dan persoalan. Universitas Ma Chung berkomitmen untuk membangun budaya pantang menyerah, tekun, tidak kenal lelah, tidak mudah putus asa dalam diri setiap civitas

4) Kreatif

Universitas Ma Chung mendorong terciptanya budaya kerja yang inovatif, produktif, dan imajinatif, sehingga senantiasa dapat mengembangkan hal-hal dan cara-cara baru dalam bekerja dan berhasil

5) Dinamis

Universitas Ma Chung berkomitmen untuk menciptakan lingkungan pekerjaan dan pembelajaran yang senantiasa hidup, bergairah, dan aktif sehingga memampukan segenap civitas akademika Universitas Ma Chung untuk mengantisipasi, mengadaptasi, dan mengakomodasi.

6) Ramah dan Menyenangkan

Universitas Ma Chung mendorong terciptanya lingkungan kerja dan pembelajaran yang tertib, penuh kegembiraan dan menyenangkan untuk menghasilkan sumber daya manusia yang ramah, toleran, pembawa damai, kegembiraan, dan sukacita.

7) Meritokratik

Universitas Ma Chuung menghargai prestasi, kerja keras, dan kontribusi nyata. Universitas Ma Chung percaya bahwa prestasi menentukan posisi.

8) Profesional

Universitas Ma Chung berkomitmen untuk membangun semangat kerja yang selalu mengedepankan kualitas dan motivasi untuk menjadi yang terbaik di dalam setiap upaya yang dilakukan.

9) Betanggung Jawab

Universitas Ma Chung mendorong segenap civitas akademikanya untuk selalu mampu mempertanggungjawabkan semua pemikiran, tindakan, dan ucapan dengan baik dan benar.

10) Sinergi

Universitas Ma Chung mendorong segenap civitas akademikanya untuk selalu mampu mempertanggungjawabkan semua pemikiran, tindakan, dan ucapan dengan baik dan benar.

11) Rendah Hati

Universitas Ma Chung mendorong segenap civitas akademikanya untuk selalu mampu mempertanggungjawabkan semua pemikiran, tindakan, dan ucapan dengan baik dan benar.

12) Citizenship

Universitas Ma Chung proaktif memberikan kontribusi dalam membangun masyarakat dan lingkungan hidup yang aman, sehat, damai, sejahtera, adil, dan makmur.

Universitas Ma Chung memiliki 10 program studi yaitu: Manajemen Bisnis, Akutansi Bisnis, Sistem Informasi Bisnis, Teknik Informatika, Desain Komunikasi Visual, Farmasi, Teknokimia Pangan, Teknik Industri, *English Letters*, *Chinese Business Language and Culture*.

Visi dari program studi Teknik Informatika Universitas Ma Chung adalah menjadi program studi teknik informatika aras utama di Indonesia timur yang mendukung eksplorasi sumber daya alam beserta pengelolaan bisnisnya sebagai perwujudan memuliakan Tuhan dan berkontribusi nyata bagi kesejahteraan masyarakat.

Program studi Teknik Informatika Universitas Ma Chung memiliki dua bidang peminatan yaitu: sistem cerdas, dan sistem komputer. Bidang peminatan sistem cerdas berfokus pada bidang kecerdasan buatan sedangkan Bidang peminatan sistem komputer berfokus pada jaringan komputer. Proyek Praktik Kerja Lapangan yang dilakukan ini termasuk kedalam peminatan sistem cerdas. Selain memberikan fasilitas tersebut, program studi Teknik Informatika Universitas Ma Chung juga memiliki kelompok riset yaitu kelompok riset *Precision Agriculture* (PA). Melalui kelompok riset tersebut, program studi Teknik Informatika Universitas Ma Chung ikut serta mewujudkan partisipasinya untuk mengembangkan agrobisnis di Indonesia melalui bidang teknik informatika.

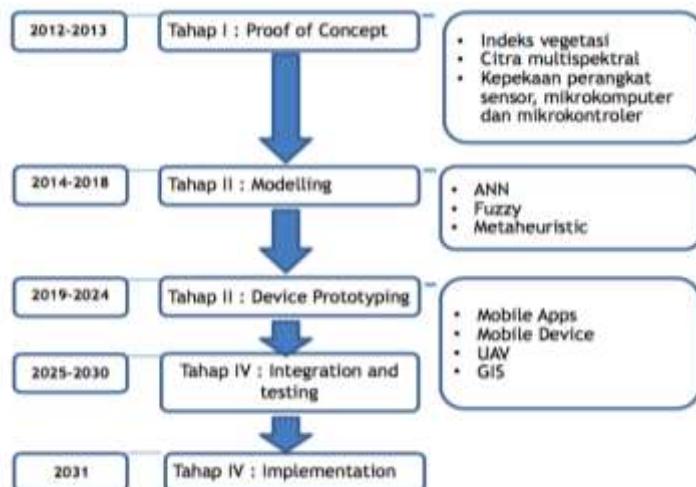
## **2.2 Roadmap Penelitian Bidang *Precision Agriculture***

Precision agriculture (PA) adalah pemanfaatan teknologi modern untuk mengumpulkan data-data kuantitatif dengan mudah dan cepat sehingga menghasilkan keputusan-keputusan yang tepat dalam pengelolaan lahan pertanian (Shafi & Mumtaz, 2019). Keputusan-keputusan tersebut terkait dengan data-data lapangan yang berpengaruh langsung maupun tidak langsung terhadap pertumbuhan tanaman. Sebagai contoh, data kadar air pada tanah dapat dijadikan dasar pengaturan waktu dan jumlah penyiraman yang tepat. (Neupane, 2019). Indonesia sebagai negara yang memiliki potensial tempat lahan yang sangat baik seharusnya juga harus bisa memaksimalkan tingkat efisiensi penanaman pada lahan tersebut sehingga dapat menghasilkan kualitas tanaman yang bagus serta meningkatkan perkembangan produk-produk agrobisnisnya.

Pada bidang penelitian *precision agriculture* Universitas Ma Chung telah berkomitmen untuk mendukung upaya-upaya pemerintah Indonesia dalam

memanfaatkan sumber daya alam hayati dengan sebaik-baiknya agar meningkatkan daya saing bangsa. Universitas Ma Chung memiliki komitmen untuk mengembangkan topik *precision agriculture* dengan tema unggulannya yaitu *Intelligent System* atau biasa dinamakan dengan sistem kecerdasan buatan. Melalui komitmen tersebut dibentuklah Kelompok Riset *Precision Agriculture* yang berfokus pada penelitian pengembangan tanaman.

Sejak tahun 2012 kelompok riset *precision agriculture* Universitas Ma Chung memiliki tahapan-tahapan penelitian yang dikerjakan dari tahun ke tahun. Secara umum tahapan-tahapan tersebut adalah pembuktian konsep yang kemudian dilakukan pemodelan, membuat prototipe, integrasi beberapa fasilitas yang sebelumnya dikembangkan secara modular, melakukan tes dan akhirnya implementasi di masyarakat melihat pada gambar 2.1



Gambar 2.1 Peta Jalan Penelitian *Precision Agriculture* di UMC

Selain *roadmap* yang ditunjukan pada gambar 2.1 kelompok riset *precision agriculture* juga memiliki rencana riset untuk 5 tahun kedepan. Secara umum terdapat tiga topik utama (pengukuran pigmen tanaman, deteksi suara unggas dan deteksi penyakit tanaman sawit). Perencaan dari ketiga topik tersebut

disusun pada tabel 2.1 yang menjelaskan rencana masing-masing dari ketiga topik tersebut selama 5 tahun kedepan.

Tabel 2.1 *Roadmap* Penelitian Kelompok Riset Precision Agriculture

No	Topik Riset	2021	2022	2023	2024	2025
1	Pengukuran Pigmen Tanaman Secara Non Destruktif		Peningkatan algoritma pengenalan warna pada daun tanaman agar invariant terhadap perbedaan cahaya dan jenis kamera			Implementasi aplikasi untuk menjadi alat ukur pigmen yang dapat digunakan pada lingkungan sesungguhnya dan bernilai komersil serta terhubung dalam sistem cloud
2	Pengenalan Suara Unggas untuk identifikasi spesies langka		Studi preprosesing data suara dan desain algoritma Machine Learning yang sesuai untuk pengenalan suara			Pembuatan aplikasi berbasis perangkat bergerak untuk deteksi suara unggas
3	Pengenalan Penyakit Tanaman Sawit melalui Citra Satelit		Studi preprosesing data citra satelit dan desain algoritma Machine Learning untuk pengenalan penyakit Sawit			Pembuatan aplikasi dan perangkat untuk deteksi perkebunan sawit yang terinfeksi penyakit

Topik riset mulai dari tahun 2021 dibagi menjadi tiga topik, yaitu: Pengukuran Pigmen Tanaman Secara Non Desktruktif, Pengenalan Suara Unggas untuk Identifikasi Spesies Langka, dan Pengenalan Penyakit Tanaman Sawit melalui Citra Satelit. Pada topik pertama, rencana penelitian terapan untuk peningkatan algoritma pengenalan warna pada daun tanaman agar invariant terhadap perbedaan cahaya dan jenis kamera akan dilakukan pada periode tahun 2021 sampai 2023 yang kemudian dilanjutkan pada penelitian pengembangan mengenai implementasi aplikasi untuk menjadi alat ukur pigmen yang dapat digunakan pada lingkungan sesungguhnya dan bernilai komersil serta terhubung dalam sistem cloud

pada periode tahun 2024 sampai 2025. Topik yang kedua, pada periode tahun 2021 sampai 2023 akan dilakukan penelitian dasar tentang studi preprosesing terhadap data suara dan desain algoritma *machine learning* yang sesuai untuk pengenalan suara. Kemudian dilanjutkan untuk penelitian terapan mengenai pembuatan aplikasi berbasis perangkat bergerak untuk deteksi suara unggas. Topik ketiga akan dilakukan penelitian dasar untuk studi preprosesing data citra satelit dan desain algoritma *machine learning* untuk pengenalan penyakit sawit pada periode tahun 2021 sampai 2022 yang selanjutnya dilanjutkan untuk penelitian terapan tentang pembuatan aplikasi dan perangkat untuk deteksi perkebunan sawit yang terinfeksi penyakit pada periode tahun 2023 sampai 2025

## **BAB III**

### **TINJAUAN PUSTAKA**

#### **3.1 Pigmen Tanaman**

Pigmen adalah zat yang mampu menyerap cahaya. Pada daun tanaman, pigmen merupakan senyawa kimia dengan ikatan rangkap terkonjugasi (kromofor) yang memiliki kemampuan untuk menyerap dan memantulkan cahaya pada panjang gelombang tertentu. Hal inilah yang memungkinkan pigmen dapat menyerap energi matahari dan sehingga dapat menampilkan warna tertentu pada daun tanaman. Pigmen juga memiliki peranan penting dalam proses metabolisme tanaman seperti pada proses fotosintesis yang dimana energi foto akan digunakan untuk mengubah air dan karbondioksida menjadi makanan (karbohidrat).

Jenis pigmen pada tanaman biasanya memiliki lebih dari satu jenis pigmen walaupun hanya akan satu yang mendominasi. Jenis pigmen pada tanaman dapat diklasifikasikan ke dalam beberapa kelompok, yaitu: klorofil, karotenoid (karoten, xantofil), flavonoid (chalcones, antosianin, flavon, flavonol) dan betalain (betaxanthin, betacyanin) (Młodzińska, 2009).

##### **3.1.1 Klorofil**

Klorofil adalah pigmen tanaman magnesium-tetraapirol yang memberi warna hijau pada tanaman, alga, dan cyanobacteria, dan merupakan pigmen utama dalam proses fotosintesis tanaman. Klorofil dan karotenoid terletak bersamaan didalam kloroplas. Pada tumbuhan tingkat tinggi klorofil terbagi menjadi dua macam yaitu *green-blue chlorophyll a*, dan *green-yellow chlorophyll b* dengan jumlah masing-masing tergantung pada spesies tumbuhan, ketersediaan mineral (Mg) dan kondisi cahaya.

Klorofil menyerap cahaya merah pada panjang gelombang 600 – 700 nm dan cahaya biru pada panjang gelombang 400 – 500 nm dengan sangat kuat. Pigmen klorofil juga memantulkan cahaya hijau pada panjang gelombang 450 – 650 nm.

Pantulan tersebutlah yang membuat tanaman memiliki warna visual hijau. Grafik spektrum absorbansi pada klorofil dapat dilihat pada gambar 3.1.

### **3.1.2 Karotenoid**

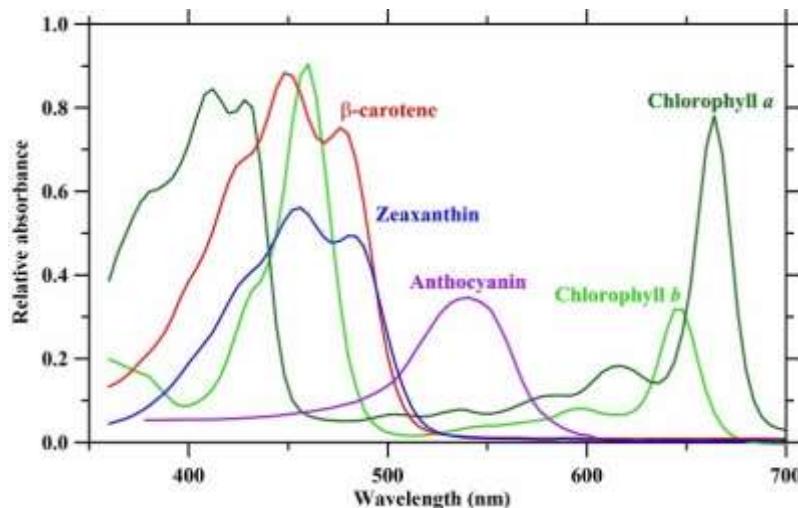
Karotenoid adalah pigmen tanaman yang memiliki sifat anti-air (*water-repellant*). Karotenoid memiliki peran pada fotosintetik dan protektif pada daun tanaman. Kadungan karotenoid pada tumbuhan memiliki jenis yang berbeda serta termasuk dalam subkelompok: karoten atau xantofil. *One carotene pigment (b-karoten)* dan *five xanthophylls* (lutein, zeaxanthin, violaxanthin, antheraxanthin dan neoxanthin) adalah jenis pigmen karotenoid pada daun yang paling umum.

Karotenoid memberikan warna visual kuning kemerahan karena menyerap kuat cahaya biru dengan panjang gelombang 400 – 500 nm dan memantulkan cahaya kuning-merah pada panjang gelombang 500 – 760 nm. Grafik spektrum absorbansi karotenoid dapat dilihat pada gambar 3.1.

### **3.1.3 Antosianin**

Antosianin memiliki visual warna merah pada tanaman. Antosianin merupakan pigmen flavonoid yang larut dalam air. Antosianin akan aktif diproduksi ketika terjadi berubahan tekanan lingkungan, proses penuaan dan ketika fotosistem belum berkembang. Antosianin akan lebih mudah terlihat pada daun ketika suatu tanaman mengalami penuaan, hal ini dapat terjadi karena tingkat fosfat yang sebelumnya tinggi tidak ada lagi (fosfat bertanggung jawab untuk pemecahan gula yang diproduksi klorofil).

Antosianin menyerap cahaya biru-hijau pada gelombang 450 – 550 nm dan memantulkan cahaya merah pada gelombang 600 nm. Grafik spektrum absorbansi pada antosianin dapat dilihat pada gambar 3.1.



Gambar 3.1 Grafik Absorbansi Pigmen Tanaman

(Samuel Eichhorn & Wu, 2019)

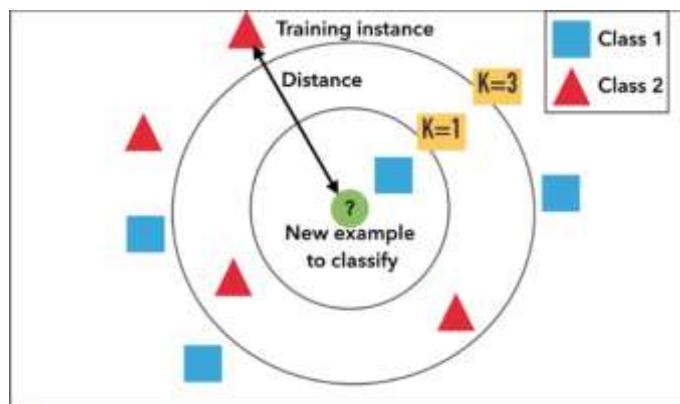
### 3.2 Color Correction

Kalibrasi citra gambar atau color correction adalah proses pengubahan kontras cahaya pada suatu gambar sehingga mendapatkan warna yang diharapkan dengan cara mengurangi atau menambahkan jumlah kontras cahaya pada titik-titik tertentu pada gambar. Kalibrasi citra gambar saat ini membantu dalam berbagai hal contohnya adalah dalam pengambilan citra di bawah laut/malam hari, filter kamera dan filter cahaya pada suatu perfilman. Warna yang ditangkap oleh kamera tidak hanya dihasilkan dari properti lingkungan tempat namun juga dari kondisi cahaya seperti pencahayaan geometri dan iluminasi warna dan juga dari karakteristik perangkat yang digunakan untuk pengambilan gambar. Tidak seperti mata manusia, perangkat kamera tidak dapat beradaptasi secara langsung terhadap tangkapan spektral dari perubahan kondisi cahaya di lingkungan yang berbeda-beda. (Gasparini & Schettini, 2014)

### 3.3 Algoritma K-nearest neighbors

Algoritma K-nearest neighbor (KNN) adalah algoritma klasifikasi non parametrik yang dikembangkan oleh Evelyn Fix dan Joseph Hodges pada tahun 1951

yang selanjutnya diperluas oleh Thomas Cover. Algoritma KNN memiliki prinsip voting untuk melakukan klasifikasi dimana dalam menentukan suatu keputusan akan dibutuhkan voting berdasarkan jumlah  $k$  (tetangga terdekat) dimana  $k$  harus berjumlah ganjil. KNN bekerja berdasarkan jarak euklid antara *test sample* dan *training sample* yang sudah ditentukan. KNN akan mencari karakteristik – karakteristik yang sama berdasarkan dari *input* yang dimasukan ( $x$ ) dan menggunakan karakteristik tersebut untuk klasifikasi hasil akhirnya ( $y$ ) (Peterson, 2009). Gambaran algoritma K-nearest neighbor dapat dilihat pada gambar 3.2.

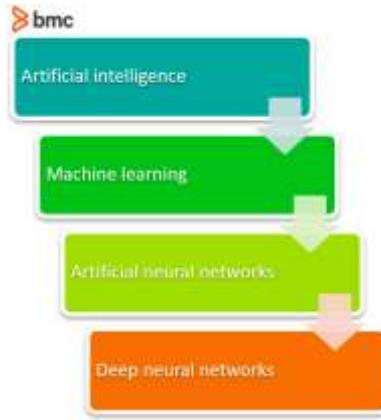


Gambar 3.2 Algoritma K-nearest neighbor

(Suliztia, 2018)

### 3.4 Deep neural network

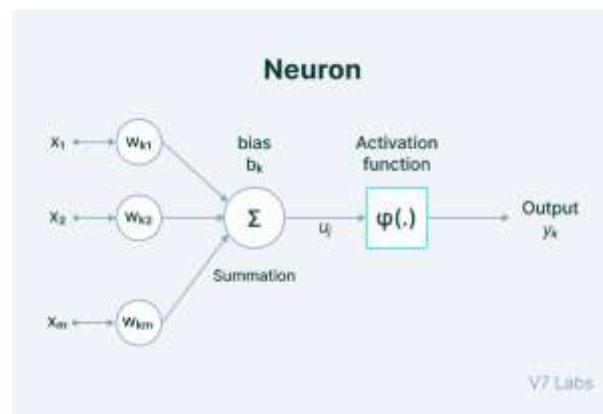
*Deep neural network* adalah sebuah rancangan *Artificial Neural Network* (ANN) yang memiliki jumlah *layer* lebih dari satu di antara *input* dan output *layernya*. Dalam dunia AI *Deep neural network* termasuk pada tahap keempat yang apabila diurutkan, yaitu: *Artificial Intelligence*, *Machine learning*, *Artificial neural networks*, dan setelahnya adalah *Deep neural networks* sesuai pada gambar 3.3



Gambar 3.3 Posisi *Deep neural network* pada dunia AI

(Johnson, 2020)

Dapat dikatakan bahwa *Deep neural networks (DNN)* adalah sekumpulan *layer neural networks* yang diperdalam atau memiliki *layer* tambahan didalamnya. *Neural networks* secara umum terdiri dari 3 jenis, yaitu: *Artificial neural network (ANN)*, *Convolutional neural network (CNN)* dan *Recurrent neural network (RNN)*. Komponen dari neuron adalah *input*, *weight*, *transfer function*, *activation function*, dan *bias* (Gambar 3.5).. Sedangkan *neural network* memiliki arsitektur yang hampir sama yaitu terdapat *input layer*, *hidden layer*, dan *output layer* (Pragati, 2021).



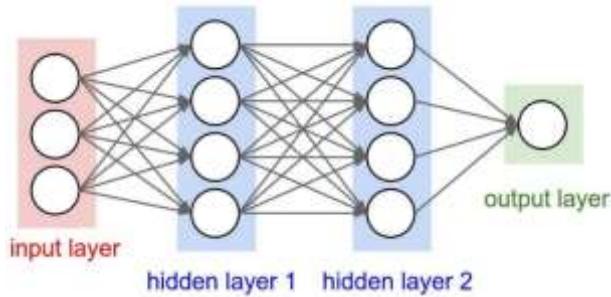
Gambar 3.4 Neuron pada *Artificial Neural Network*

(Pragati, 2021)

Komponen *input* adalah komponen yang mengolah data untuk dimasukkan kedalam model. *Weight* adalah komponen yang fungsi utamanya memberikan arti bagi fitur-fitur yang secara langsung berkontribusi untuk proses *learning*. Pada komponen ini pula proses *backpropagation* dilakukan. *Backpropagation* adalah metode *tuning weights* pada arsitektur neural berdasarkan dari *error rate* yang didapatkan pada *epoch* sebelumnya (Daniel, 2021). Selain hal tersebut, model neural juga memiliki *optimizer* yang berguna untuk membantu mengurangi *loss* saat proses *learning*. Penggabungan sekumpulan *input* menjadi satu *output* adalah tugas dari *transfer function*. Penggabungan tersebut dilakukan dengan cara menghitung total keseluruhan dari *input* menggunakan *transfer function*. *Activation function* adalah komponen yang mengenalkan *non-linearity* di dalam perceptron untuk mempertimbangkan variasi linearitas dengan *input*. *Bias* memiliki peranan untuk menggeser nilai yang dihasil oleh *activation function* (Gambar 3.6).

Setiap *layer* pada sistem neural memiliki *node* yang mempengaruhi proses belajar dari DNN. *Input layer* dan *output layer* secara umum memiliki 1 *node* sedangkan untuk *hidden layer* *node* yang diberikan dapat dirubah sesuai dengan kebutuhan pembuat model. *Input layer* adalah lapisan tempat data dimasukkan untuk diproses oleh model. *Input* yang dimaksud dapat berupa bentuk file csv atau *array* yang mewakili *pixel* pada suatu citra. Data biasanya diolah terlebih dahulu sebelum dimasukan kedalam lapisan *input* dengan tujuan agar hasil fitur yang didapatkan tepat dan tidak mengganggu proses *learning*. *Hidden layer* adalah lapisan yang membuat *deep learning* sekarang. *Hidden layer* adalah lapisan yang bertugas mengkomputasi dan mengekstrasi semua fitur yang terdapat di dalam data. Lapisan ini dapat terbagi menjadi 2 atau lebih *hidden layer* yang saling berhubungan, tujuannya adalah untuk mencari fitur tersembunyi yang terdapat di dalam data. Sebagai contoh pada pengolahan gambar, *hidden layer* pertama digunakan untuk pengambilan fitur seperti tepi atau bentuk sedangkan untuk *hidden layer* lainnya akan digunakan untuk mengidentifikasi suatu objek. *Output layer* adalah lapisan yang menerima hasil akhir dari *input layer* yang sudah diolah oleh *hidden layer*. Lapisan ini juga akan memberikan hasil prediksi berdasarkan dari model yang sudah melakukan proses *learning*. *Output layer* secara umum terdiri

dari satu *node* saja. Satu *hidden layer* dapat dinamakan ANN sedangkan apabila memiliki dua atau lebih *hidden layer* maka arsitektur tersebut dinamakan DNN. Arsitektur DNN dapat dilihat pada gambar 3.6.



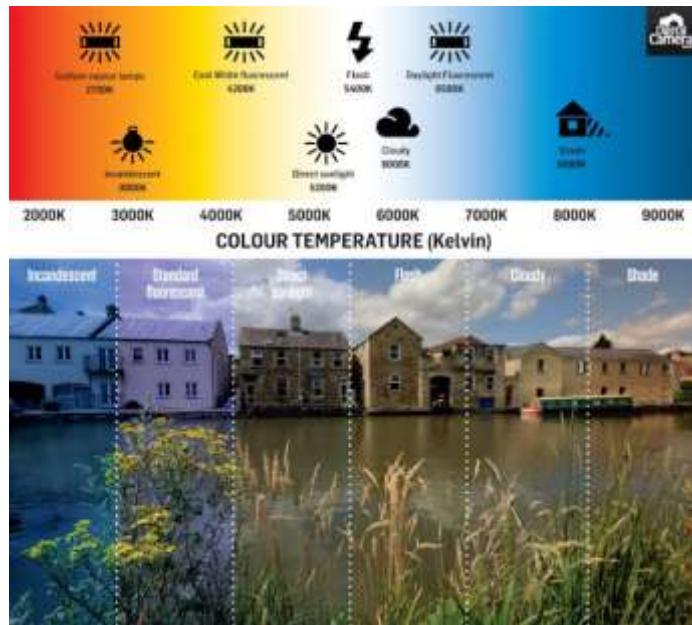
Gambar 3.5 Arsitektur *Deep neural networks*

(Johnson, 2020)

### 3.5 White-Balance

White balance adalah teknik yang digunakan untuk menyeimbangkan warna putih pada suatu citra gambar. Dalam proses penyeimbangan warna putih tersebut citra akan ditambahkan lawan warnanya dari sumber cahaya tujuannya untuk mendapatkan warna netral sedangkan untuk warna putih akan tetap terlihat putih. Untuk mendapatkan warna netral pada saat *white balancing*, warna dasar yang digunakan adalah *Red, Green, Blue* (RGB). Warna *Red* (Merah) berlawanan dengan warna *Cyan* (Biru Langit), warna *Green* (Hijau) berlawanan dengan warna *Magenta* (Ungu Magenta), dan warna *Blue* (Biru) berlawanan dengan warna *Yellow* (Kuning). Jadi semisal suatu citra gambar memiliki warna yang terlalu kebiruan (foto dalam air) maka untuk menetralkannya adalah dengan menambahkan warna kuning sebaliknya jika warna citra terlalu kekuningan (warna sore hari) maka warna biru perlu ditambahkan untuk menetralkan warna kuning tersebut. (Mahmoud, et al., 2019). *White-balance* banyak digunakan ketika gambar yang ditangkap memiliki pencahayaan yang kurang baik. Sebagai contoh adalah ketika gambar ditangkap pada kondisi ruangan gelap, di dalam air dan kondisi pencahayaan dari sinar matahari (cahaya matahari ketika pagi dan sore). Kondisi cahaya yang

berbeda-beda akan sangat mempengaruhi hasil dari gambar, namun melalui penggunaan *white-balance* gambar yang ditangkap akan disesuaikan sehingga hasil pencahayaan terlihat normal.



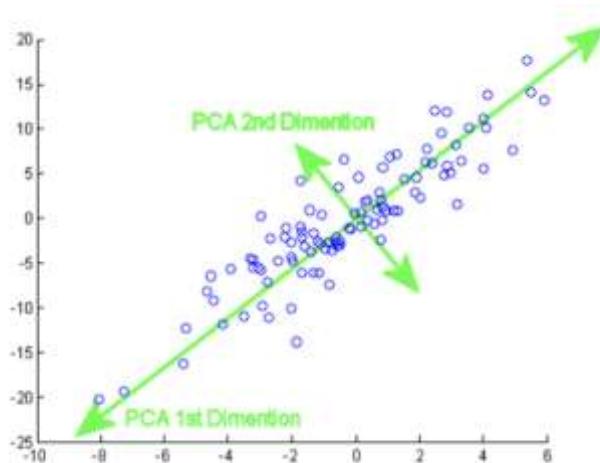
Gambar 3.6 *White Balance*

(rys, 2018)

### 3.6 Principal Component Analysis (PCA)

Principal Component Analysis adalah teknik untuk mereduksi suatu dimensi yang biasanya digunakan untuk mereduksi dimensi dari suatu dataset yang besar dengan cara metransformasi sejumlah variabel besar menjadi kecil namun tetap menyimpan sumber informasi yang dimiliki. Mengurangi dimensi suatu variabel biasanya memiliki konsekuensi pada akurasi yang dihasilkan namun tujuan asli dari reduksi dimensi adalah untuk menyederhanakan kumpulan variabel atau dimensi sebagai ganti berkurangnya akurasi dikarenakan data yang sudah direduksi akan mempermudah proses visualisasi dan analisa suatu data sehingga menjad lebih cepat. Urutan dari proses PCA sendiri terdiri dari 5 langkah umum yaitu pertama menstandarisasi data tujuannya adalah agar data memiliki nilai yang sama ketika

digabungkan dengan PCA langkah kedua adalah menghitung komputasi matriks kovarian tujuannya untuk mencari relasi antar variabel dikarenakan beberapa variabel terkadang memiliki korelasi yang tinggi dan memiliki informasi yang hampir sama, langkah ketiga adalah menentukan nilai *eigenvector* dan *eigenvalues* pada matriks kovarian untuk mengidentifikasi komponen *pricipal*, Langkah keempat adalah membuat fitur vektor untuk menentukan komponen *principal* mana saja yang harus disimpan, langkah kelima adalah melakukan perombakan data berdasarkan sumbu di komponen *principalnya* hal ini dapat dilakukan dengan mengkalikan hasil fitur yang di *tranpose* dengan hasil standarisasi dataset yang sudah di *transpose* juga (Jaadi, 2021).



Gambar 3.7 Ilustrasi PCA

(Seb, 2021)

### 3.7 Arsitektur Multi-*Decoder*

Multi *Decoder* adalah istilah yang digunakan untuk bentuk arsitektur CNN yang memiliki dua atau lebih arsitektur *decoder*. *Encoder* adalah *layer* yang menerima *input* dan meencodenya menjadi bentuk representasi yang sudah ditentukan sedangkan *decoder* adalah *layer* yang menerima hasil dari *encoder* dan

menggunakannya untuk melakukan prediksi dan menggenerasi hasilnya *outputnya* (Afifi & Michael Brown, 2020).

### **3.8 Pengolahan Citra Digital**

Pengolahan citra digital adalah pengolahan suatu citra dengan menggunakan perangkat digital menggunakan algoritma-algoritma yang ada. Citra adalah merupakan fungsi dari intensitas cahaya yang merepresentasikan suatu gambar foto atau bidang dua dimensi lainnya. Pengolahan citra digital tidak hanya menggunakan satu algoritma saja dalam prosesnya namun juga dapat mengaplikasikan algoritma-algoritma lain sehingga mendapatkan suatu hasil citra yang diharapkan (Bevans & Chattman, 2004). Salah satu contoh permasalahan yang dapat diselesaikan menggunakan algoritma citra digital adalah ketika pengambilan gambar di luar angkasa. Tanpa dilakukan pengolahan citra, gambar di luar angkasa tidak akan memiliki warna yang dapat dilihat oleh mata manusia melainkan hanya akan berupa gambar hitam dan putih saja atau bahkan berwarna lain. Melalui pengolahan citra digital, gambar tersebut dapat diolah sehingga warna yang dimasukan dapat diterima oleh mata manusia. Contoh lainnya adalah pengolahan citra digital pada foto hitam putih. Sebelum muncul kamera yang dapat mengambil warna RGB, foto dengan hasil warna hitam putih terlebih dikembangkan oleh manusia. Sekarang dengan menggunakan pengolahan citra digital foto tersebut dapat diubah sehingga warna yang sebelumnya hitam putih dapat divisualisasikan dengan warna RGB.

### **3.9 Python**

*Python* adalah bahasa pemrograman multifungsi yang dikembangkan oleh Guido van Rossum dan kemudian diperkenalkan secara publik pada tahun 1991. Python adalah salah satu bahasa pemrograman yang dapat melakukan sejumlah intruksi multi guna secara langsung (interpretatif) dengan metode orientasi objek (*Object Oriented Programming*) yang menggunakan semantik dinamis untuk memberikan tingkat keterbacaan *syntax*. *Python* memberikan fasilitas untuk banyak

paradigma pemrograman seperti struktural (khusunya secara prosedur), orientasi objek dan *functional programming*.

### **3.10 Color Transformation**

*Color transformation* atau transformasi warna adalah transformasi suatu representasi warna dari satu ruang warna ke ruang warna lainnya. Kalkulasi ini akan bekerja ketika terjadi suatu penukaran data didalamnya. Transformasi dilakukan dengan cara mengambil referensi profil data kedalam sebuah standart ruang warna. Hal ini akan memudahkan warna untuk berubah ke warna lain yang terdapat pada referensi standart ruang warna, selain itu juga harus dipastikan bahwa referensi standart ruang warna harus melingkupi seluruh warna yang dapat dilihat oleh mata manusia.

### **3.11 Github Algoritma KNN dan DNN**

Github adalah suatu tempat untuk *developer* dan penggunanya untuk saling berbagi proyek dalam bentuk situs *website*. Selain sebagai tempat berbagi github juga digunakan untuk mengerjakan suatu proyek secara bersama-sama melalui sistem *repositories*, *branches*, *commits*, dan *pull requests* yang dimiliki oleh github. Situs tersebut juga menyediakan referensi-referensi untuk pelajar dan peniliti yang salah satu contohnya adalah proyek kalibrasi warna.

Kalibrasi warna yang didapatkan dari github adalah kalibrasi WB sRGB dan *Deep White-balance* yang keduanya dibagikan oleh peneliti yang sama yaitu Mahmoud Nafifi. Proyek yang dibagikan berupa kode aplikasi, dataset dan jurnal. Selain hal tersebut Mahmoud Nafifi juga menyediakan contoh-contoh dari hasil kalibrasi dalam bentuk gambar (Afifi & Price, 2019) dan (Afifi & Brown, 2020).

### 3.12 Ringkasan Jurnal Tentang *When Color Constancy Goes Wrong*

Pada jurnal ini menjelaskan tentang kegagalan suatu filter white-balance ketika melakukan tugasnya yaitu *white-balancing*. Kegagalan dalam penggunaan white-balance menyebabkan gambar dengan kontras cahaya gelap kebiruan menjadi terlalu terang dan gambar dengan kontras cahaya yang terlalu terang menjadi gelap salah satu contohnya dapat dilihat pada gambar 3.8. Pada gambar tersebut ditunjukan seperti apakah gambar yang gagal dilakukan *white-balance* (*Images with incorrect WB applied*). Selain itu gambar tersebut juga menunjukan bagaimana hasil perbaikan dengan menggunakan *diagonal correction*, koreksi dengan linerisasi dan koreksi dengan algoritma KNN. Sedangkan untuk gambar terakhir paling kanan adalah gambar yang seharusnya benar ketika dilakukan *white-balancing*, hal tersebut dikenal dengan istilah gambar *ground-truth*.



Gambar 3.8 Perbedaan Penggunaan *White-balance*

(Mahmoud, et al., 2019)

Melalui penggunaan algoritma K-nearest neighbor maka suatu kegagalan white-balance dapat diperbaiki. Secara berurutan, prosedur pertama yang dilakukan adalah dengan cara mempersiapkan dataset *training* yang akan digunakan oleh

algoritma KNN. Data *training* ini nantinya akan dibentuk menjadi suatu histogram yang menfiturkan *log-chrominance space*. *Log-chrominance space* adalah representasi dari distribusi warna dari suatu gambar  $\mathbf{I}$  dengan tensor  $m \times m \times 3$  ( $3m^2$ ) yang parameternya adalah  $uv$  ( $u$  = horizontal,  $v$  = vertical). Hal tersebut dinamakan dengan histogram RGB- $uv$ . Nilai dari histogram tersebut dihasilkan dari fungsi  $h(\mathbf{I})$  yang dideskripsikan dengan perhitungan berikut:

$$\begin{aligned}
 \mathbf{I}_{y(i)} &= \sqrt{\mathbf{I}_{R(i)}^2 + \mathbf{I}_{G(i)}^2 + \mathbf{I}_{B(i)}^2}, \\
 \mathbf{I}_{u1(i)} &= \log(\mathbf{I}_{R(i)}) - \log(\mathbf{I}_{G(i)}), \\
 \mathbf{I}_{v1(i)} &= \log(\mathbf{I}_{R(i)}) - \log(\mathbf{I}_{B(i)}), \\
 \mathbf{I}_{u2} &= -\mathbf{I}_{u1}, \mathbf{I}_{v2} = -\mathbf{I}_{u1} + \mathbf{I}_{v1}, \\
 \mathbf{I}_{u3} &= -\mathbf{I}_{v1}, \mathbf{I}_{v2} = -\mathbf{I}_{u1} + \mathbf{I}_{u1}, \\
 \mathbf{H}(\mathbf{I})_{(u,v,C)} &= \sum_i \mathbf{I}_{y(i)} \left[ |\mathbf{I}_{uC(i)} - u| \leq \frac{\varepsilon}{2} \wedge |\mathbf{I}_{vC(i)} - v| \leq \frac{\varepsilon}{2} \right], \\
 h(\mathbf{I})_{(u,v,C)} &= \sqrt{\frac{\mathbf{H}(\mathbf{I})_{(u,v,C)}}{\sum_{u'} \sum_{v'} \mathbf{H}(\mathbf{I})_{(u',v',C)}}},
 \end{aligned} \tag{3-1}$$

dimana  $i = \{1, \dots, N\}$ , R, G, B merepresentasikan warna pada gambar  $\mathbf{I}$ , dan  $C \in \{1, 2, 3\}$  merepresentasikan tiap warna yang terdapat pada histogram dan  $\varepsilon$  adalah lebar bin dari histogram. Menghitung akar dari nilai  $\mathbf{H}$  yang sudah dinormalisasi akan meningkatkan kemampuan diskriminatif pada fitur histogram yang diproyeksikan.

Demi menambah efisiensi, maka langkah selanjutnya adalah dengan cara mengaplikasikan fitur PCA pada hasil dari histogram. Perhitungan dari PCA direpresentasikan sebagai berikut:

$$v(\mathbf{I}) = \mathbf{W}^T (\text{vec}(h(\mathbf{I})) - \mathbf{b}), \tag{3-2}$$

Dimana nilai  $u(\mathbf{I}) \in R^c$  adalah fitur PCA yang berisikan komponen  $c$  pada *principal component* (PC). Nilai  $c$  pada PCA ini berjumlah 55 komponen. Nilai dari PCA ini nantilah yang akan digunakan oleh KNN untuk mencari nilai  $k$  pada PCA yang memiliki karakteristik sama dengan data *input*. Proses dari *input* data sendiri juga hampir sama, yaitu dengan melalui histogram-*uv* dan kemudian dilakukan PCA. Hasil PCA dari gambar *input* akan digunakan untuk mencari gambar dengan karakteristik yang sama menggunakan algoritma KNN. Setelah gambar sudah ditemukan, maka langkah selanjutnya adalah menghitung nilai *weight* menggunakan kombinasi linear dari matriks yang dikoreksi  $M_s$  sebagai berikut:

$$\mathbf{M} = \sum_{j=1}^k a_j \mathbf{M}_s^{(j)}, \quad (3-3)$$

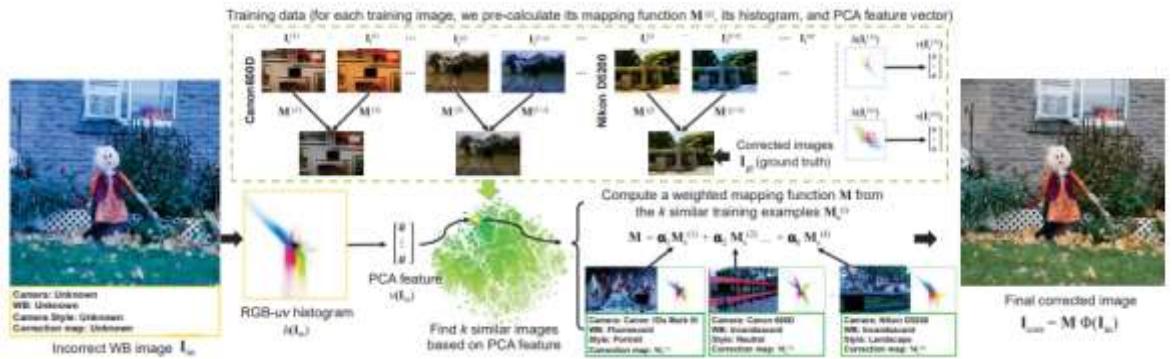
Dimana nilai  $a$  adalah nilai *weight* vektor yang merepresentasikan fungsi radial:

$$a_j = \frac{\exp(-\frac{d_j^2}{2\sigma^2})}{\sum_{k'=1}^k \exp(-\frac{d_{k'}^2}{2\sigma^2})}, j \in [1, \dots, k], \quad (3-4)$$

$\sigma$  adalah faktor nilai jatuh radial dan  $\mathbf{d}$  merepresentasikan vektor yang berisikan jarak  $L_2$  antara fitur *input* yang dimasukan dengan gambar  $k$  yang didapatkan pada dataset *training* saat melakukan pencarian KNN. Pada hasil akhir, gambar koreksi *white-balance* yang dihasilkan didapatkan dari asosiasi dengan data *training*. Sehingga rumus perhitungan pada gambar akhir adalah sebagai berikut:

$$\mathbf{I}_{corr} = \mathbf{M}\Phi(\mathbf{I}_{in}). \quad (3-5)$$

dengan nilai  $\mathbf{M}$  adalah proses mapping pada perhitungan *weight* sebelumnya (Mahmoud, et al., 2019).



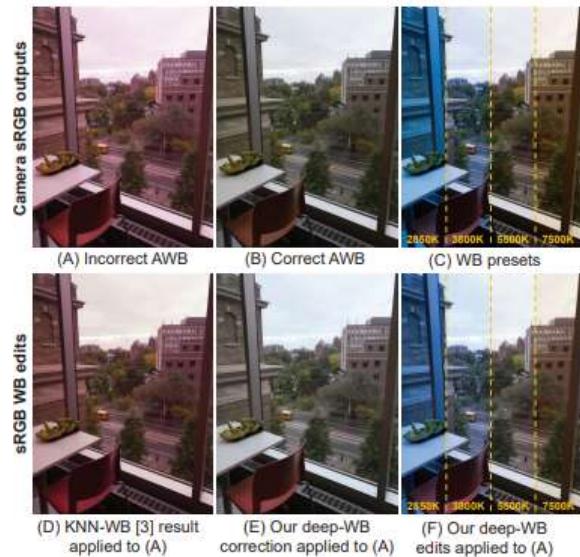
Gambar 3.9 Diagram Prosedur Algoritma KNN

(Mahmoud, et al., 2019)

### 3.13 Ringkasan Jurnal Tentang Deep White-Balance

Jurnal *Deep White-Balance Editing* memiliki tujuan yang sama dengan jurnal “*When Color Constancy Goes Wrong*” yaitu untuk memperbaiki gambar yang mengalami kegagalan dalam melakukan *white balancing*. Penulis beranggapan bahwa dengan menggunakan *Deep neural network* gambar yang gagal dalam proses *white-balance* dapat diperbaiki dengan lebih baik dan lebih akurat. Arsitektur *Deep neural network* yang digunakan terdiri dari kerangka *multi-decoder* untuk sutingan *sRGB White-balance* yang didalamnya terdiri dari satu *encoder* dan dua *decoder*. *Decoder* pertama adalah *Auto WB decoder* dan *decoder* keduanya adalah *Shade WB decoder*. *Auto WB decoder* akan menghasilkan gambar yang lebih keputihan karena prosesnya sama yaitu *white-balancing* (WB) sedangkan untuk *Shade WB decoder* akan menghasilkan gambar yang terlihat lebih hangat. Proses *training* dilakukan dengan cara memilih secara acak gambar yang terdapat pada dataset *Rendered WB*. Melalui hal tersebut gambar *testing* akan menghasilkan suatu pengaturan target WB dengan menggunakan hasil *training decoder*. Gambar 3.10 merepresentasikan perbedaan hasil antara penggunaan algoritma KNN untuk *white-balance* dan *deep*

*learning* untuk *white-balance*. Gambar A adalah gambar yang mengalami kegagalan *white-balance*, gambar B adalah gambar yang seharusnya benar ketika dilakukan *white-balance*, gambar C adalah hasil gambar dengan temperatur warna yang berbeda-beda dengan satuan *kelvin* (K), gambar D adalah hasil dari algoritma KNN ketika dimasukan *input* dari gambar A, gambar E adalah hasil dari algoritma *deep learning* ketika dimasukan *input* gambar A, dan yang terakhir gambar F adalah hasil dari algoritma *deep learning* ketika dimasukan *input* gambar A dan diolah dengan temperatur warna yang berbeda-beda (Mahmoud & Michael, 2020).



Gambar 3.10 Perbedaan Penggunaan Algoritma KNN dan DNN

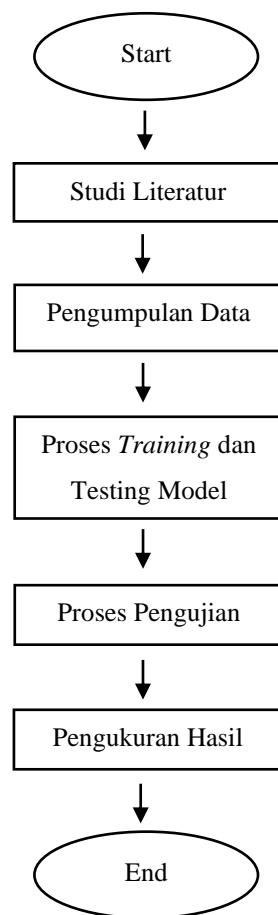
(Mahmoud & Michael, 2020)

## **BAB IV**

### **HASIL DAN PEMBAHASAN**

#### **4.1 Proses Pengerjaan**

Proyek dari Praktik Kerja Lapangan ini bertujuan untuk mengimplementasikan teknik kalibrasi warna pada *leaf piction* agar dapat digunakan untuk mengambil foto citra tanpa terpengaruh kondisi cahaya yang terdapat di lapangan. Tahapan pengerjaan yang digunakan seperti pada gambar 4.1 berikut.



Gambar 4.1 Tahapan Pengerjaan

Tahapan penggerjaan dilakukan dengan pencarian referensi-referensi jurnal tentang kalibrasi warna tanpa harus melakukan segmentasi, mencari referensi kode tentang kalibrasi dan dataset yang akan digunakan. Tahap selanjutnya adalah melakukan pengolahan data yang sudah didapatkan agar dapat digunakan untuk proses kalibrasi, setelah data sudah diolah maka tahap selanjutnya adalah penjelasan tentang arsitektur model dan kemudian melakukan uji coba. Pada tahap terakhir akan dilakukan pengukuran hasil dan dilihat perbandingan yang didapatkan untuk menentukan apakah proses kalibrasi berhasil dilakukan atau tidak

## 4.2 Studi Literatur

Pada tahap studi literatur ini adalah penjelasan mengenai bagaimana referensi-referensi didapatkan dari jurnal-jurnal artikel hingga kode dari lama github. Pencarian literatur pertama kali dilakukan dengan membaca jurnal dan artikel mengenai kalibrasi warna secara *unsupervised*. Melalui artikel-artikel tersebut didapatkan bahwa penulis sering menggunakan metode *white-balance* untuk algoritma *unsupervised* yang mereka kerjakan. Setelah mengetahui tentang kinerja *white-balance*, tahap selanjutnya adalah mencari artikel atau situs github yang menjelaskan proses *white-balance* secara spesifik terutama ketika artikel tersebut dilatar belakangi tentang permasalahan pada suatu kondisi cahaya dan diselesaikan menggunakan suatu algoritma *artificial intelligence* (AI).

Dalam pencarian tersebut didapatkan sebuah laman github yang memberikan referensi berupa jurnal bersertakan kode untuk aplikasinya. Referensi pertama yang didapatkan adalah tentang pembedahan *white-balance* kamera menggunakan algoritma *K-nearest neighbor* dengan penulis utamanya adalah Mahmoud Afifi. Setelah itu didapatkan pula kode aplikasi dalam bentuk *python* yang dijalankan melalui *jupyter notebook*. Setelah referensi pertama didapatkan, selanjutnya adalah melakukan pencarian tentang penulis dari jurnal tersebut. Berdasarkan hasil pencarian kedua didapatkan pula referensi lanjutan mengenai pembedahan *white-balance* namun kali ini menggunakan algoritma *deep learning*.

Kode aplikasi pada algoritma *deep learning* ini adalah hasil konversi dari ekstensi *python* yaitu *pytorch*.

#### **4.2.1 Pengumpulan Data Referensi**

Pada proses pengumpulan data referensi utama yang diambil adalah jurnal mengenai pemberahan *white-balance* (Mahmoud, et al., 2019). Secara jurnal ini menjelaskan tentang bagaimana suatu kamera dengan pengaturan *white-balance* gagal dalam melakukan pengolahan gambarnya sehingga hasil yang dimunculkan tidak sesuai dengan harapan. Selain memberikan referensi jurnal, penulis juga memberikan referensi berupa kode aplikasi yang digunakan melalui situs *github*. Referensi jurnal lainnya adalah jurnal dengan judul “*Deep White-Balance Editing*”. Jurnal ini menjelaskan permasalahan yang sama yaitu tentang teknik untuk memperbaiki *white-balance* yang gagal namun dengan algoritma *Deep neural network*. Pada jurnal ini penulis menjabarkan tentang bagaimana algoritma Deep Learning lebih akurat dalam melakukan *white-balancing* dengan cara menggunakan 2 tipe *decoder* untuk melakukan proses *training* agar mendapatkan hasil yang lebih baik.

#### **4.2.2 Arsitektur Model**

Pada bagian ini adalah penjelasan mengenai arsitektur model dari masing-masing algoritma dan bagaimana arsitektur tersebut membantu dalam mengatasi permasalahan *white-balance* beserta penjelasan mengenai fungsi dan fitur-fitur yang digunakan pada arsitektur

##### **4.2.2.1 Arsitektur Model KNN**

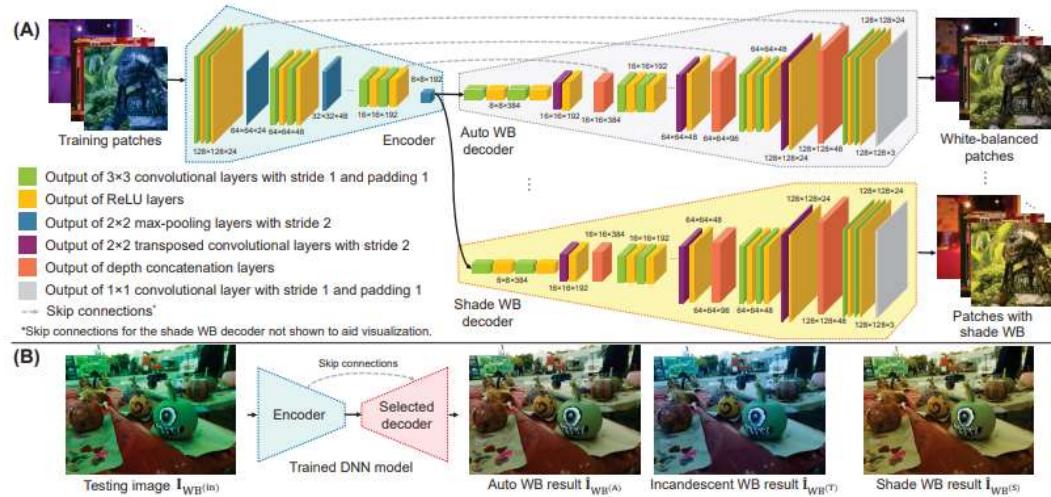
Arsitektur model KNN yang dibentuk terdiri dari 2 model arsitektur. Model yang pertama menggunakan dataset *training* untuk set 1 dimana dataset tersebut diambil menggunakan kamera DSLR selain itu nilai *k* yang digunakan untuk model pertama adalah 25. Model yang kedua menggunakan dataset *training* untuk set 1

dan juga set 2 dimana set 2 memiliki beberapa gambar yang ditangkap menggunakan kamera handphone, nilai  $k$  yang dimiliki untuk model kedua adalah 75. Selanjutnya masing-masing model memiliki nilai sigma senilai 0.25 (faktor *fall-off* untuk KNN *blending*). Arsitektur KNN juga menggunakan gamut mapping yaitu *clipping*. Proses algoritma KNN gambar 3.10, menjelaskan bahwa algoritma KNN digunakan setelah gambar yang dimasukan sebagai *input* dihitung nilai RGB-uv histogramnya dan dilakukan PCA. Hasil PCA nantinya akan digunakan untuk mencari nilai gambar pada dataset *training* yang memiliki nilai PCA sama. Pencarian pada dataset *training* menggunakan nilai  $k$  yang sudah ditentukan pada model. Setelah gambar-gambar pada dataset *training* ditemukan maka tahap selanjutnya adalah menghitung bobot (*weight*) pada masing-masing gambar dari dataset *training* untuk mendapatkan nilai yang sesuai agar gambar *input* dapat dikalibrasi dengan benar. Masing-masing set memiliki variabel dependen dan independen pada algoritma KNN yang sama yaitu dataset pada *training* sebagai independen variabelnya ( $x$ ) dan *input* citra sebagai independen variabelnya ( $y$ ).

#### 4.2.2.2 Arsitektur Model *Deep neural network*

Arsitektur model pada algoritma *Deep neural network* terdiri dari rangkaian *multi-decoder* yang terdiri dari satu *encoder* dan dua *decoder*. *Encoder* terdiri dari 3 *layer convolutional* dan 3 *layer max pooling*. Berdasarkan gambar 4.2 *layer* berwarna hijau adalah *layer convolutional* 3x3 dengan stride 1 dan padding 1 dengan menggunakan fungsi ReLU (*layer* berwarna kuning). Sedangkan untuk *layer* berwarna biru adalah *layer max pooling* 2x2 dengan stride bernilai 2. *Decoder* yang digunakan terdapat 2 macam *decoder* yaitu *Auto WB decoder* dan *Shade WB decoder*. *Auto WB decoder* adalah pengaturan *white-balance* agar gambar yang didapatkan lebih keputih-putihan yang disesuaikan berdasarkan temperatur warna pada *input* gambar. *Auto WB* memiliki 10 *layer* yang terdiri dari 4 *layer convolutional* 3x3, 3 *layer transposed convolutional* 2x2 dengan stride 2 (*layer* berwarna ungu) dan 3 *layer depth concatenation* (*layer* berwarna oranye). *Shade WB decoder* adalah pengaturan *white-balance* untuk memberikan warna yang lebih

hangat. Arsitektur *decoder* dari *Shade WB* memiliki bentuk arsitektur yang sama dengan *Auto WB*. Jika melihat pada gambar 4.2 terdapat hasil untuk *Incandescent WB* dimana hasil tersebut akan memberikan gambar seperti kondisi cahaya didalam ruangan.



Gambar 4.2 Arsitektur Model DNN

(Mahmoud, et al., 2019)

### 4.3 Pengumpulan Dataset

Masing-masing algoritma memiliki dataset yang berbeda. Pada proses ini akan dijelaskan tentang dataset yang digunakan oleh masing-masing algoritma, cara generasi dataset dan pengolahannya hingga dapat digunakan untuk proses *training* serta dataset yang digunakan untuk proyek kalibrasi warna sendiri.

#### 4.3.1 Dataset pada Algoritma KNN

Dataset pada algoritma ini didapatkan dari hasil penangkapan citra gambar yang sudah digenerasi sehingga menghasilkan gambar dengan kondisi cahaya yang berbeda-beda dengan total hasil gambarnya adalah 65,000 gambar sRGB. Pada setiap gambar sRGB yang sudah digenerasi akan memiliki nilai *white-balance* yang

sudah sesuai atau dinamakan dengan “*ground-truth*”. Gambar terbagi menjadi 3 set yaitu: set pertama adalah dataset dengan nama *rendered wb dataset set 1* dengan cara pengambilan gambar menggunakan kamera DSLR dan besertakan *color chart* yang terdapat di lokasi sebagai referensi untuk hasil iluminasi *ground-truth*.Dataset tersebut juga berisikan beberapa gambar tanaman yang diambil di suatu kebun namun untuk gambar tentang spesifik daun itu sendiri tidak terdapat di dalam dataset. Keseluruhan jumlah datanya adalah 62,535 gambar. Set kedua dinamakan dengan *rendered WB dataset set 2* yang diambil dengan menggunakan kamera DSLR dan empat kamera handphone (iPhone 7, Google Pixel, LG G4, dan Galaxy S6 Edge) dengan total keseluruhannya berjumlah 2,881 gambar. Set yang ketiga dinamakan *cube++ dataset* yang berisikan 10,242 gambar dengan beberapa dataset berisikan gambar yang diambil di dalam ruangan dan lingkungan dengan kontras cahaya gelap. Format gambar yang ditangkap berupa raw-RGB yang nantinya dapat dirubah kedalam bentuk keluaran sRGB dimana merepresentasikan emulasi dari masing-masing pengaturan WB dan *style* gambar pada kamera.Gambar yang sudah dirubah kedalam bentuk sRGB nantinya akan diolah kedalam bentuk WB yang salah sebagai dataset. Proses untuk mendapatkan gambar yang sesuai atau benar adalah dengan cara mengambil nilai “*ground-truth*” putih secara manual dari warna abu-abu tengah pada *color rendition chart*. Setelah data *training* selesai dibentuk selanjutnya adalah menambahkan variasi untuk dataset gambar *incorrect WB* dengan cara melakukan transformasi warna. Teknik transformasi warna yang dilakukan adalah dengan menggunakan fungsi polynomial kernel yang ditemukan oleh Hong.

Tabel 4.1 Contoh Dataset yang Digunakan pada KNN dan DNN

Dataset	Incorrect WB	<i>Ground-truth</i>
Rendered WB dataset (Set 1)		
Rendered WB dataset (Set 2)		
Cube++ dataset		

### 4.3.2 Dataset pada Algoritma *Deep neural network*

Algoritma *Deep neural network* menggunakan dataset yang sama dengan algoritma KNN. Hal yang membuat perbedaan adalah pengolahan dataset tersebut untuk dijadikan data *training*. Dataset yang digunakan untuk algoritma ini terbagi menjadi dua model seperti yang dijelaskan pada arsitektur model KNN. Model pertama akan menggunakan dataset *rendered WB dataset set 1* dan model yang kedua akan menggunakan *rendered WB dataset set 2* ditambahkan dengan *cube++ dataset*. Data yang akan digunakan sebagai *training* akan terbagi menjadi 3 partisi, partisi pertama dan kedua berisikan gambar acak berjumlah 12,000 gambar dari data di set satu dan partisi yang ketiga akan digunakan untuk testing dengan jumlah

data adalah 21,046 gambar. Data *training* juga akan dilakukan augmentasi untuk masing-masing pemanndangan yang berbeda. Augmentasi data ini memiliki tambahan dimana gambar diberikan *color temperature* (suhu warna) yang berbeda-beda. Pemilihan temperatur warna ini dilakukan secara acak antara 2850K sampai 7500K. Jumlah data augmentasi yang ditambahkan untuk proses training *Deep neural network* ini berjumlah 1,029 gambar. Proses pembuatan dataset augmentasi dapat dilihat pada gambar 4.3.

#### 4.3.3 Dataset Pengujian

Dataset untuk pengujian kalibrasi diperoleh dari ibu Dr. Kestrilia Rega Prilianti, M.Si di kelompok riset *precision agriculture*. Dataset berisikan gambar daun dengan 2 jenis spesies yaitu *Duranta Erecta* dan *Piper Betle*. Masing-masing spesies memiliki jumlah 100 data gambar untuk *duranta erecta* dan 80 gambar untuk *piper betler* dengan total keseluruhan dataset adalah 180 data gambar. Gambar daun terdiri dari 10 daun yang berbeda dengan spesies sama untuk *duranta erecta* dan 8 daun untuk spesies *piper betle*. Setiap dataset memiliki karakteristik pengambilan gambar yang sama yaitu difoto dengan cahaya dan jarak kamera yang berbeda-beda. Karakteristik cahaya yang dimiliki adalah rendah, sedang, tinggi, sangat rendah dan sangat tinggi sedangkan untuk jarak kamera adalah dekat dan jauh saja. Tingkat pencahayaan pada daun *duranta erecta* dapat dilihat pada tabel 4.2.

Tabel 4.2 Tingkat Pencahayaan Daun *Duranta Erecta*

Tingkat Pencahayaan	Lux		
	Minimal	Average	Maximal
Sangat Rendah	35	40	49
Rendah	25	145	156
Sedang	502	559	568
Tinggi	1889	2610	2691
Sangat Tinggi	1737	3450	3870

Terdapat masing-masing sepuluh daun *Duranta Erecta* yang masing-masing memiliki 5 tingkat pencahayaan dan 2 jenis jarak pengambilan gambar. Jarak pengambilan gambar yang digunakan adalah 15 cm (jarak dekat) dan 40 cm (jarak jauh). Contoh dataset yang digunakan adalah daun dengan kode DE 1 (tabel 4.3).

Tabel 4.3 Contoh Dataset *Duranta Erecta* (DE 1)

Tingkat Pencahayaan	Jarak	
	Dekat	Jauh
Rendah		
Sedang		
Tinggi		
Sangat Rendah		

Tabel 4.3 (Lanjutan)

Sangat Tinggi	
---------------	--

Jarak yang digunakan untuk pengambilan dataset *Piper Betle* sama yaitu 15 cm (jarak dekat) dan 40 cm (jarak jauh). Sedangkan untuk tingkat pencahayaannya dapat dilihat pada tabel 4.4. Pengambilan gambar dilakukan pada hari yang sama dengan berpindah posisi saja. Contoh dataset yang digunakan adalah dataset *Piper Betle* dengan kode PB 1 (tabel 4.5).

Tabel 4.4 Tingkat Pencahayaan Daun *Piper Betle*

Tingkat Pencahayaan	Lux		
	Minimal	Average	Maximal
Sangat Rendah	0	43	48
Rendah	0	102	120
Sedang	0	660	668
Tinggi	0	2090	2945
Sangat Tinggi	0	46900	53521

Pada tingkat pencahayaan daun spesies *piper betle* dilakukan perubahan dimana tingkat pencahayaan minimalnya adalah 0 baik bagi seluruh kategori. Sedangkan untuk pencahayaan maksimalnya juga diperkecil dimana secara berurutan untuk kategori cahaya sangat rendah nilai maksimal pencahayaannya adalah 48, kategori rendah adalah 120, kategori sedang adalah 668, kategori tinggi adalah 2955 dan kategori sangat tinggi adalah 53521 lux. Sehingga pada hasil rata-ratanya juga berbeda (*average*) bila dibandingkan dengan tingkat pencahayaan untuk daun spesies *duranta erecta*.

Tabel 4.5 Contoh Dataset Piper Betle

Tingkat Pencahayaan	Jarak	
	Dekat	Jauh
Rendah		
Sedang		
Tinggi		
Sangat Rendah		
Sangat Tinggi		

## 4.4 Proses *Training* dan *Testing* Model

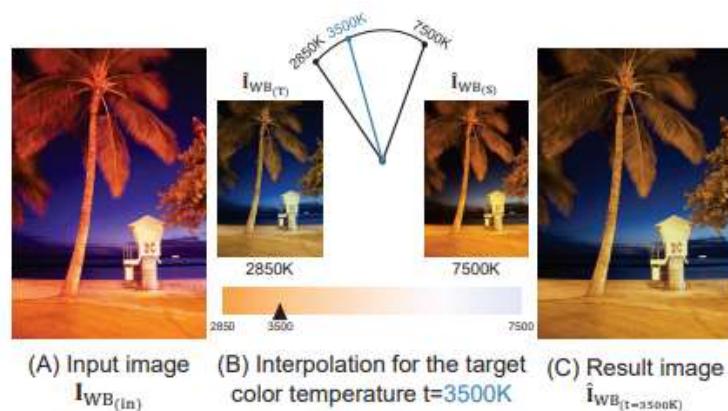
Pada proses *training* dan *testing* model akan dijelaskan bagaimana algoritma tersebut di *training* dari masing-masing algoritma sehingga dapat menghasilkan aplikasi untuk perbaikan *white-balance*.

### 4.4.1 Proses *Training* pada Algoritma KNN

Dataset *training* pada algoritma KNN akan digunakan untuk proses pencarian gambar menggunakan KNN ketika suatu *input* dimasukkan. Lalu langkah selanjutnya adalah menghitung nilai bobot seperti pada gambar 3.9.

### 4.4.2 Proses *Training* pada Algoritma *Deep neural network*

Proses *training* dataset pada algoritma *Deep neural network* pertama-tama dilakukan dengan cara menambah dataset *training* dengan cara augmentasi gambar. Augmentasi gambar dilakukan dengan cara mengubah temperatur warna untuk setiap gambar yang terdapat pada dataset *training* tujuannya adalah untuk memberikan variasi baru terhadap kondisi warna yang berbeda-beda. Pengubahan temperatur ini dilakukan secara acak untuk setiap dataset dan diberikan referensi *ground-truth* untuk hasil augmentasi dari masing-masing gambar (gambar 4.3).



Gambar 4.3 Augmentasi pada Temperatur Warna

Parameter *training* yang digunakan adalah 165,000 iterasi dengan *optimizer* Adam. Nilai *decay rate* berdasarkan pergerakan gradien senilai 0,9 dan *decay rate* untuk rata-rata pergerakan gradien bernilai 0,999 sedangkan untuk parameter *learning rate* adalah 0,5 untuk setiap 25 epoch dengan 32 batch setiap iterasi. Proses *testing* dilakukan dengan membandingkan warna gambar menggunakan fungsi *color mapping*. Teknik *color mapping* yang dilakukan adalah *polynomial color mapping*.

## 4.5 Proses Pengujian

Pada tahap pengujian akan dijelaskan bagaimana dataset daun *duranta erecta* dan *piper betle* dikalibrasi sehingga menghasilkan gambar yang telah diperbaiki melalui *white-balance* kedua algoritma.

### 4.5.1 Pemilihan dan Pengolahan Dataset *Input*

Pada proses ini akan dilakukan pemilihan pada dataset *duranta erecta* dan *piper betle* yang akan digunakan sebagai *input* masukan. Dataset yang akan digunakan adalah gambar daun nomor satu dari kedua spesies (DE 1 dan PB 1) dengan seluruh pencahayaan kecuali pencahayaan sangat tinggi dan sangat rendah. Tidak ada alasan tertentu kenapa daun nomor satu digunakan untuk pengujian, pemilihan tersebut hanya sebagai contoh saja. Daun yang digunakan sebagai pengujian akan dilakukan proses *resizing*. Proses *resize* berbeda untuk kedua algoritma. Algoritma KNN akan melakukan *resize* pada gambar jika dimensi yang dimiliki melebihi  $450 \times 450$  *pixel*.

Algoritma *Deep neural network* akan melakukan *resize input image* kebentuk ukuran maksimumnya yaitu  $656 \times 656$  *pixel* tanpa mempedulikan apakah ukuran *input* lebih kecil atau lebih besar dari ukuran yang sudah ditentukan.

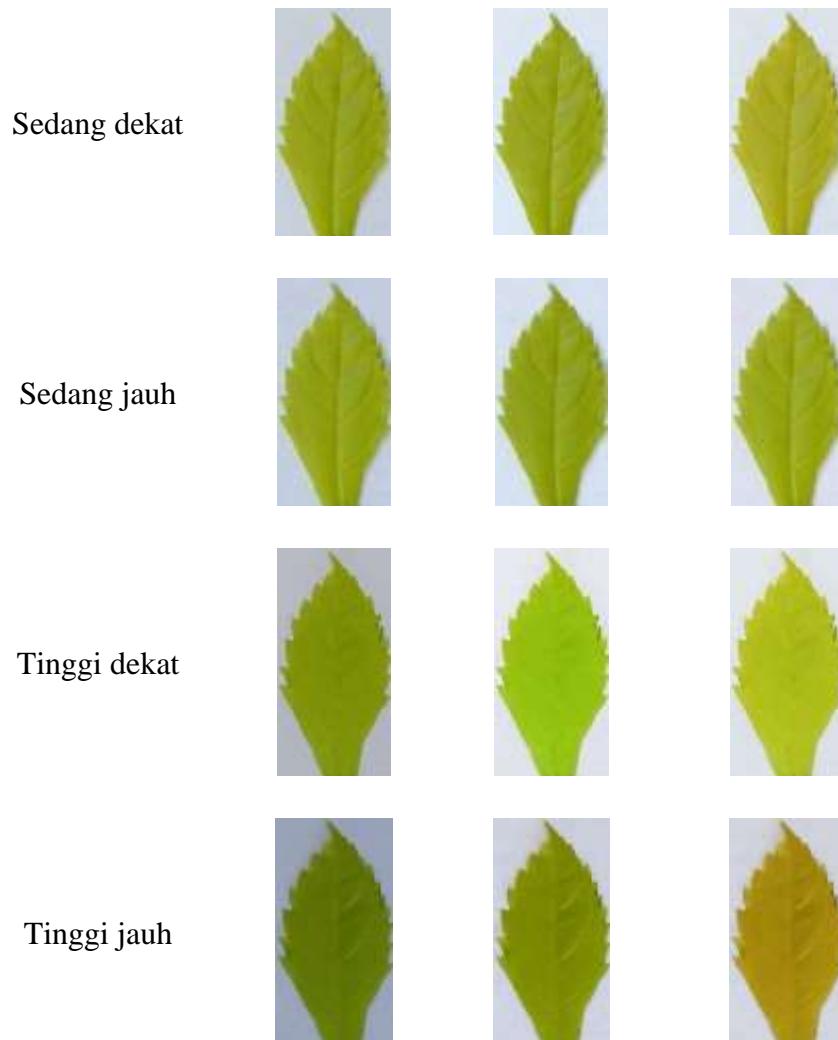
#### 4.5.2 Pengujian Dataset *Input*

Dikarenakan masing-masing dari algoritma memiliki dua model (DNN memiliki dua *decoder*). Pengujian akan dilakukan tiga kali yaitu untuk algoritma KNN yang memiliki 2 model dengan dataset yang berbeda serta yang ketiga adalah algoritma DNN untuk *decoder Auto WB* dikarenakan hasil yang dibutuhkan adalah untuk mendapatkan gambar yang memiliki warna cahaya normal yaitu dengan melakukan *Auto WB*. Tidak seperti *Auto WB*, *Shade WB* akan memberikan warna pada gambar seperti temperatur cahaya didalam ruangan (Kebiru-biruan). Dataset yang pertama kali akan diuji coba adalah *duranta erecta* dan kemudian dilanjutkan untuk spesies daun *piper betle*. Pengujian pertama akan dilakukan untuk algoritma KNN dengan kedua model (Tabel 4.6).

Tabel 4.6 Hasil Kalibrasi Algoritma KNN *Duranta Erecta* (DE 1)

Kondisi Cahaya dan Jarak	Input	Output Set 1 ( $k = 25$ )	Output Set 2 (Dataset+, $k=75$ )
Rendah dekat			
Rendah Jauh			

Tabel 4.6 (Lanjutan)



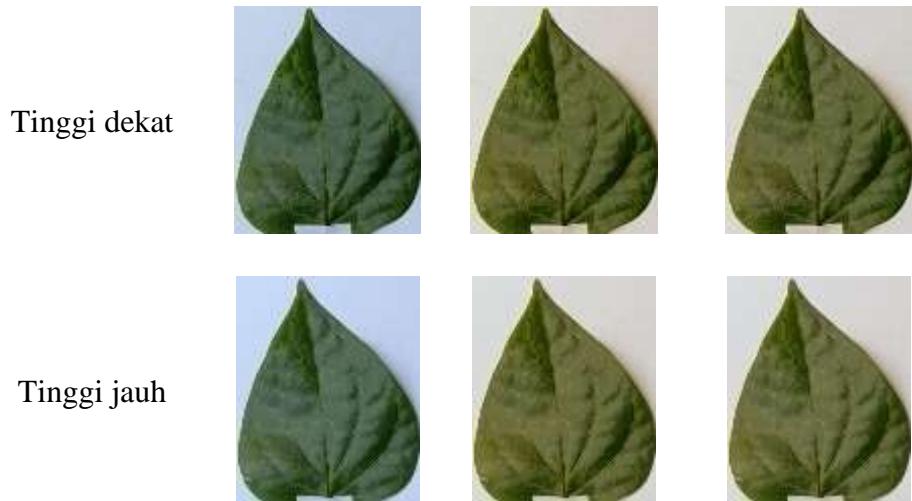
Hasil kalibrasi KNN untuk daun *duranta erecta* dapat dilihat perbedaannya mulai dari kondisi cahaya dan jarak rendah dekat, pada daun tersebut hasil set 1 hampir tidak terlihat perbedaannya sedangkan untuk set 2 perbedaannya terlihat dari daun yang menjadi sedikit keabu-abuan. Selanjutnya pada daun rendah jauh juga sama set 1 hampir tidak terlihat perbedaannya sedangkan untuk set 2 menjadi sedikit keabu-abuan. Pada daun sedang dekat hasil dari kedua set terlihat perbedaanya dimana set 1 daun terlihat lebih cerah dan kehijuan sedangkan set 2 daun sedikit berwarna merah-oranye. Pada daun sedang jauh hasil dari set 1 hampir tidak kelihatan selain *background* yang semakin putih sedangkan untuk set 2 juga

sama. Daun tinggi dekat memberikan hasil kalibrasi yang terlihat berbeda untuk set 1 dimana warna daun menjadi kehijauan sedangkan untuk set 2 daun terlihat lebih cerah. Terakhir, pada daun tinggi jauh hasil kalibrasi set 1 memberikan cahaya yang terlihat lebih jelas dan cerah sedangkan untuk set 2 daun menjadi berwarna merah-oranye. Selanjutnya adalah kalibrasi untuk daun *piper betle* (tabel 4.7).

Tabel 4.7 Hasil Kalibrasi Algoritma KNN *Piper Betle* (PB 1)

Kondisi Cahaya dan Jarak	Input	Output Set 1	Output Set 2
		(k = 25)	(Dataset+, k=75)
Rendah dekat			
Rendah jauh			
Sedang dekat			
Sedang jauh			

Tabel 4.7 (Lanjutan)



Hasil kalibrasi pada daun *piper betle* bila dilihat dari awal, daun rendah dekat menghasilkan warna cahaya yang lebih jelas dan terang baik untuk set 1 maupun set 2. Pada daun rendah jauh set 1 memberikan hasil yang juga lebih terang sedangkan pada set 2 hasilnya sedikit terang namun bila dilihat lebih teliti warna daun juga ikut berubah kecoklatan. Daun sedang dekat memberikan hasil kalibrasi yang lebih terang juga baik dari kedua set. Selanjutnya daun sedang jauh juga menghasilkan gambar yang lebih terang dan jelas dari kedua set. Daun tinggi dekat dan tinggi jauh memberikan hasil kalibrasi yang lumayan mirip dimana daun menjadi lebih jelas namun warnanya juga ikut berubah sedikit kecoklatan. Sebelumnya percobaan untuk mengganti nilai  $k$  sudah pernah dilakukan namun secara visual gambar yang dihasilkan sama saja.

Proses kalibrasi selanjutnya yang dilakukan adalah kalibrasi menggunakan algoritma *Deep Neural Networks* (DNN), secara berurutan proses pengujiannya akan dilakukan untuk daun spesies *duranta erecta* terlebih dahulu yang kemudian dilanjutkan untuk daun spesies *piper betle*. Seperti yang sudah dijelaskan sebelumnya, proses pengujian kalibrasi hanya akan dilakukan untuk *decoder Auto WB* saja dikarenakan tujuan dari proyek ini adalah mendapatkan gambar dengan pencahayaan normal menggunakan *white balancing*.

Tabel 4.8 Hasil Kalibrasi Algoritma DNN *Duranta Erecta* (DE 1)

Nama Daun	Input	Output DNN
Rendah dekat		
Rendah Jauh		
Sedang dekat		
Sedang jauh		
Tinggi dekat		

Tabel 4.8 (Lanjutan)

Tinggi jauh



Pada Tabel 4.8, gambar daun *duranta erecta* yang dihasilkan oleh kalibrasi DNN memberikan warna yang lebih jelas bila dibandingkan dengan hasil kalibrasi KNN. Namun daun dengan pencahayaan tinggi dan kamera jauh memberikan hasil kalibrasi yang terlalu jauh sehingga warna daun juga ikut berubah. Hal ini dapat disebabkan karena algoritma berusaha mengubah warna daun dengan menambah warna sebaliknya yang terlalu besar sehingga menyebabkan hasil kalibrasi membuat daun menjadi lebih tua (terlalu banyak penambahan warna abu-abu).

Tabel 4.9 Hasil Kalibrasi Algoritma DNN *Piper Betle* (PB 1)

Nama Daun

*Input*

Ouput DNN

Rendah dekat



Rendah Jauh



Tabel 4.9 (Lanjutan)



Tabel 4.9 adalah hasil kalibrasi untuk daun dengan spesies *piper betle*. Hasil yang didapatkan dari kalibrasi daun *piper betle* membuat pencahayaan terlihat lebih natural dan sama dengan gambar daun lainnya. Hasil kalibrasi pada daun *piper betle* menghasilkan gambar yang tidak *over*-kalibrasi sehingga pencahayaan dan warna daun yang dihasilkan terlihat lebih baik bila dibandingkan dengan spesies daun *duranta erecta*.

## 4.6 Pengukuran Hasil

Pada tahap ini akan dijelaskan bagaimana gambar hasil kalibrasi di hitung secara kuantitatif untuk melihat keefektifan dari kedua algoritma. Pengukuran hasil yang dilakukan akan dibagi menjadi dua bagian yaitu pengukuran hasil berdasarkan kesamaan gambar dan pengukuran hasil berdasarkan kesamaan jumlah pigmen pada daun.

### 4.6.1 Pengukuran Berdasarkan Kesamaan Gambar

Pengukuran hasil secara kuantitatif dilakukan dengan cara menghitung kemiripan gambar yang sudah dikalibrasi dengan menggunakan perhitung RMSE dan SSIM. Perhitungan yang dilakukan menggunakan fungsi *image similarity* yang dimiliki pada *library* python. Metode perbandingan yang dilakukan adalah dengan cara membandingkan nilai setiap *pixel* yang terdapat pada gambar menggunakan teknik *color mapping* (Kirshi, 2021). Hasil perhitungan untuk algoritma KNN dan DNN beserta keterangan variabel adalah sebagai berikut.

Cal\_1 = rendah jauh,

Cal\_2 = rendah dekat,

Cal\_3 = sedang jauh,

Cal\_4 = sedang dekat,

Cal\_5 = tinggi jauh,

Cal\_6 = tinggi dekat.

Tabel 4.10 Hasil *Similarity Original Duranta Erecta*

	RMSE					
	Cal_1	Cal_2	Cal_3	Cal_4	Cal_5	Cal_6
Cal_1	-	-	-	-	-	-
Cal_2	0.00527	-	-	-	-	-
Cal_3	0.00512	0.00386	-	-	-	-
Cal_4	0.00603	0.00434	0.00349	-	-	-
Cal_5	0.01233	0.01162	0.01022	0.01114	-	-
Cal_6	0.00754	0.00644	0.00514	0.00614	0.00642	-

Tabel 4.10 (Lanjutan)

	SSIM					
	Cal_1	Cal_2	Cal_3	Cal_4	Cal_5	Cal_6
Cal_1	-	-	-	-	-	-
Cal_2	0.98247	-	-	-	-	-
Cal_3	0.97114	0.97942	-	-	-	-
Cal_4	0.96506	0.98002	0.98948	-	-	-
Cal_5	0.89159	0.9113	0.93199	0.92317	-	-
Cal_6	0.93507	0.95422	0.96889	0.96179	0.97713	-

Tabel 4.10 adalah hasil uji *similarity* untuk daun *duranta erecta*. Jika dilihat dari hasil uji untuk daun *duranta erecta* tanpa kalibrasi menghasilkan nilai yang relatif mendekati. Kemudian penjelasan variabel dapat dilihat pada tabel disebelah kanan.

Tabel 4.11 Hasil *Similiarity Original Piper Betle*

	RMSE					
	Cal_1	Cal_2	Cal_3	Cal_4	Cal_5	Cal_6
Cal_1	-	-	-	-	-	-
Cal_2	0.00461	-	-	-	-	-
Cal_3	0.00605	0.00615	-	-	-	-
Cal_4	0.00831	0.00727	0.00757	-	-	-
Cal_5	0.01061	0.01131	0.01004	0.01419	-	-
Cal_6	0.00893	0.00951	0.00931	0.01175	0.00823	-

	SSIM					
	Cal_1	Cal_2	Cal_3	Cal_4	Cal_5	Cal_6
Cal_1	-	-	-	-	-	-
Cal_2	0.96772	-	-	-	-	-
Cal_3	0.93943	0.93527	-	-	-	-
Cal_4	0.86909	0.92934	0.92704	-	-	-
Cal_5	0.83633	0.83476	0.86613	0.78656	-	-
Cal_6	0.86909	0.86924	0.88937	0.83149	0.95189	-

Tabel 4.11 adalah hasil uji *similarity* untuk daun *piper betle* tanpa kalibrasi. Pada hasil uji ini, terdapat beberapa nilai perbandingan yang cukup berbeda yaitu cal\_4 dengan cal\_5 dimana nilainya adalah 0.786.

Tabel 4.12 Hasil *Similarity* KNN Model Set 1 *Duranta Erecta*

	RMSE					
	Cal_1	Cal_2	Cal_3	Cal_4	Cal_5	Cal_6
Cal_1	-	-	-	-	-	-
Cal_2	0.00585	-	-	-	-	-
Cal_3	0.00543	0.00511	-	-	-	-
Cal_4	0.00676	0.00447	0.00436	-	-	-
Cal_5	0.00653	0.00782	0.0058	0.0084	-	-
Cal_6	0.00687	0.00606	0.00539	0.00607	0.00814	-

	SSIM					
	Cal_1	Cal_2	Cal_3	Cal_4	Cal_5	Cal_6
Cal_1	-	-	-	-	-	-
Cal_2	0.98316	-	-	-	-	-
Cal_3	0.968	0.97154	-	-	-	-
Cal_4	0.96584	0.97696	0.98553	-	-	-
Cal_5	0.91707	0.91822	0.95243	0.92904	-	-
Cal_6	0.95717	0.96544	0.97769	0.96731	0.96029	-

Tabel 4.12 adalah hasil *similarity* daun *duranta erecta* dengan kalibrasi KNN set 1. Bila dibandingkan dengan hasil tanpa kalibrasinya, terdapat perubahan pada beberapa nilai walaupun tidak begitu signifikan. Hal ini dapat dengan mudah dilihat pada tabel SSIM dimana nilainya saling mendekati di antara 0.9. Bila dibandingkan dengan hasil visualnya, hasil perubahan kalibrasi memang terlihat sangat jelas namun bila dihitung secara kuantitatif menggunakan uji *similarity* nilai yang dihasilkan justru tidak begitu signifikan.

Pengujian yang selanjutnya dilakukan adalah uji *similarity* untuk hasil kalibrasi dengan algoritma KNN set 2. Pengujian yang dilakukan sama yaitu dengan cara uji *similarity* dengan parameter hasilnya berupa RMSE dan SSIM.

Tabel 4.13 Hasil *Similarity* KNN Model Set 2 *Duranta Erecta*

	RMSE					
	Cal_1	Cal_2	Cal_3	Cal_4	Cal_5	Cal_6
Cal_1	-	-	-	-	-	-
Cal_2	0.00531	-	-	-	-	-
Cal_3	0.0072	0.00632	-	-	-	-
Cal_4	0.00781	0.00571	0.00434	-	-	-
Cal_5	0.00769	0.00789	0.00567	0.00748	-	-
Cal_6	0.00863	0.00761	0.00498	0.00516	0.0073	-

	SSIM					
	Cal_1	Cal_2	Cal_3	Cal_4	Cal_5	Cal_6
Cal_1	-	-	-	-	-	-
Cal_2	0.98773	-	-	-	-	-
Cal_3	0.93184	0.9416	-	-	-	-
Cal_4	0.95084	0.96484	0.98232	-	-	-
Cal_5	0.90289	0.90886	0.96982	0.94861	-	-
Cal_6	0.93641	0.94741	0.97985	0.97864	0.96702	-

Tabel 4.13 adalah hasil uji *similarity* daun *duranta erecta* dengan kalibrasi KNN set 2. Nilai yang dihasilkan pada tabel ini juga menghasilkan perubahan walaupun tidak signifikan. Bila dibandingkan dengan hasil tanpa kalibrasi, nilai SSIM yang dihasilkan pada tabel ini juga saling mendekati di antara 0.9. Namun jika dicermati kalibrasi KNN set 1 menunjukkan hasil yang lebih baik dibandingkan dengan set 2. Secara keseluruhan hasil uji *similarity* pada spesies daun *duranta erecta* menunjukkan perubahan namun tidak terlalu signifikan atau kecil sekali. Bila dilihat dari perkembangan hasil kedua algoritma, DNN menunjukkan hasil yang lebih baik dibandingkan dengan algoritma KNN baik untuk set satu maupun set kedua.

Pengujian yang selanjutnya dilakukan adalah untuk spesies daun *piper betle*. Urutan pengujian yang dilakukan sama dengan spesies daun *duranta erecta* yaitu uji *similarity* untuk hasil kalibrasi algoritma KNN set 1 dan set 2 kemudian dilanjutkan untuk hasil kalibrasi algoritma DNN.

Tabel 4.14 Hasil *Similarity* KNN Model Set 1 *Piper Betle*

	RMSE					
	Cal_1	Cal_2	Cal_3	Cal_4	Cal_5	Cal_6
Cal_1	-	-	-	-	-	-
Cal_2	0.00482	-	-	-	-	-
Cal_3	0.00674	0.00686	-	-	-	-
Cal_4	0.00824	0.00817	0.00796	-	-	-
Cal_5	0.01017	0.0088	0.0085	0.0112	-	-
Cal_6	0.00922	0.00813	0.00908	0.0096	0.00806	-

	SSIM					
	Cal_1	Cal_2	Cal_3	Cal_4	Cal_5	Cal_6
Cal_1	-	-	-	-	-	-
Cal_2	0.96471	-	-	-	-	-
Cal_3	0.93655	0.93868	-	-	-	-
Cal_4	0.92065	0.9252	0.92897	-	-	-
Cal_5	0.85747	0.9052	0.90713	0.8579	-	-
Cal_6	0.87895	0.91976	0.91323	0.88551	0.95392	-

Pada tabel 4.14 nilai yang dihasilkan dari kalibrasi KNN set 1 untuk daun *piper betle* menunjukkan perubahan yang tidak terlalu signifikan pula. Bila dibandingkan dengan hasil tanpa kalibrasi, nilai SSIM yang dihasilkan lebih mendekati pada nilai 0.89.

Tabel 4.15 Hasil *Similarity* KNN Model Set 2 *Piper Betle*

	RMSE					
	Cal_1	Cal_2	Cal_3	Cal_4	Cal_5	Cal_6
Cal_1	-	-	-	-	-	-
Cal_2	0.00464	-	-	-	-	-
Cal_3	0.00694	0.00698	-	-	-	-
Cal_4	0.00817	0.0081	0.00798	-	-	-
Cal_5	0.00935	0.00915	0.00839	0.01124	-	-
Cal_6	0.00837	0.00835	0.00892	0.00957	0.008	-

Tabel 4.15 (Lanjutan)

	SSIM					
	Cal_1	Cal_2	Cal_3	Cal_4	Cal_5	Cal_6
Cal_1	-	-	-	-	-	-
Cal_2	0.97506	-	-	-	-	-
Cal_3	0.94416	0.94006	-	-	-	-
Cal_4	0.92934	0.9265	0.93055	-	-	-
Cal_5	0.88876	0.90658	0.91165	0.86311	-	-
Cal_6	0.90852	0.91907	0.91686	0.88964	0.95479	-

Tabel 4.15 adalah hasil kalibrasi KNN set 2 untuk daun *piper betle* ini menunjukkan nilai yang lebih baik apabila dibandingkan dari citra tanpa kalibrasi dan kalibrasi KNN set 1. Hal ini dapat dilihat pada hasil tabel SSIM yang dihasilkan mulai saling mendekata di antara 0.91.

Tabel 4.16 Hasil *Similarity DNN Duranta Erecta*

	RMSE					
	Cal_1	Cal_2	Cal_3	Cal_4	Cal_5	Cal_6
Cal_1	-	-	-	-	-	-
Cal_2	0.00507	-	-	-	-	-
Cal_3	0.97994	0.00508	-	-	-	-
Cal_4	0.98285	0.00584	0.00348	-	-	-
Cal_5	0.88759	0.01437	0.01415	0.01405	-	-
Cal_6	0.97192	0.00767	0.00634	0.00559	0.012	-

	SSIM					
	Cal_1	Cal_2	Cal_3	Cal_4	Cal_5	Cal_6
Cal_1	-	-	-	-	-	-
Cal_2	0.98531	-	-	-	-	-
Cal_3	0.97994	0.9734	-	-	-	-
Cal_4	0.98285	0.9712	0.98979	-	-	-
Cal_5	0.97192	0.889	0.87252	0.86902	-	-
Cal_6	0.88759	0.96211	0.97083	0.97469	0.88191	-

Selanjutnya pada tabel 4.16 adalah hasil uji *similarity duranta erecta* dengan kalibrasi DNN. Secara keseluruhan nilai SSIMnya sudah mulai mendekati angka 1 namun masih terdapat beberapa yang bahkan berkurang. Hal ini dapat disebabkan karena hasil kalibrasi yang sudah ditunjukkan pada tabel 4.9 terdapat gambar yang *over-calibrated*. Bila dibandingkan dengan KNN dan tanpa kalibrasi, algoritma DNN sudah cukup baik dalam menyamakan kondisi pencahayaan hanya saja beberapa gambar masih perlu diperhatikan agar tidak terjadi *over-calibrated*.

Gambar yang terjadi *over-calibrated* biasanya menghasilkan warna yang terlalu berbeda dari tujuan awal white balance, hal ini dapat disebabkan karena warna atau kontras cahaya yang terdapat pada gambar terlalu tinggi sehingga mengakibatkan *white-balance* menambahkan lawan warna yang terlalu berlebihan. Selanjutnya uji DNN untuk gambar daun *piper betle* akan dilakukan dan dianalisis hasilnya apakah lebih baik daripada algoritma KNN.

Tabel 4.17 Hasil *Similarity* DNN *Piper Betle*

	RMSE					
	Cal_1	Cal_2	Cal_3	Cal_4	Cal_5	Cal_6
Cal_1	-	-	-	-	-	-
Cal_2	0.00383	-	-	-	-	-
Cal_3	0.00558	0.00571	-	-	-	-
Cal_4	0.00784	0.00706	0.00704	-	-	-
Cal_5	0.00942	0.01042	0.00968	0.0134	-	-
Cal_6	0.00781	0.00869	0.00899	0.011	0.008	-

	SSIM					
	Cal_1	Cal_2	Cal_3	Cal_4	Cal_5	Cal_6
Cal_1	-	-	-	-	-	-
Cal_2	0.97862	-	-	-	-	-
Cal_3	0.95353	0.94606	-	-	-	-
Cal_4	0.93323	0.93849	0.93848	-	-	-
Cal_5	0.8917	0.85917	0.87751	0.81812	-	-
Cal_6	0.91789	0.89131	0.90025	0.85975	0.9525	-

Tabel 4.17 adalah hasil kalibrasi DNN daun *piper betle*. Pada hasil tersebut nilai yang dihasilkan sudah saling mendekati namun masih tidak sebaik algoritma KNN. Hal ini sama permasalahannya dengan kalibrasi DNN untuk daun *duranta erecta* dimana terdapat beberapa gambar yang *over-calibrated* sehingga menyebabkan beberapa gambar menjadi terlalu tua atau berbeda dari gambar aslinya.

Secara keseluruhan hasil uji *similarity* yang dihasilkan hampir tidak menghasilkan nilai yang terlalu signifikan dimana perbedaan kuantifikasinya hanya menambah atau berkurang sedikit saja. Perubahan kecil ini bisa saja terjadi karena dataset yang digunakan untuk kalibrasi tidak terfokus pada satu objek saja, melainkan berupa gambar-gambar pemandangan dan dalam ruangan. Gambar-gambar tersebut melakukan *white-balance* dengan melihat kondisi lingkungan di sekitarnya juga dimana dipengaruhi oleh banyak objek dan faktor seperti warna ruangan dan rak buku jika dataset pada dataset *indoor*. Sedangkan pada dataset *outdoor* faktor yang dapat mempengaruhi adalah seperti taman bunga, langit dan dedaunan di tempat pengambilan gambar. Sebelumnya percobaan untuk mengganti nilai  $k$  pada algoritma KNN juga dihitung uji *similarity*nya namun nilai yang dihasilkan hanya berubah sangat kecil sekali bahkan terdapat beberapa hasil yang tidak menunjukkan perubahan.

#### 4.6.2 Pengukuran Berdasarkan Kesamaan Hasil Pigmen

Pengukuran yang kedua dilakukan berdasarkan persamaan jumlah pigmen pada tumbuhan yang sama. Pengukuran hasil yang dilakukan menggunakan model prediksi dari *leaf piction* yaitu model P3NET. Nilai perbandingan yang digunakan adalah standart deviasi dengan cara menghitungnya untuk setiap jenis pencahayaan dan jarak. Semakin kecil nilai standart deviasi dari prediksi pigmen maka semakin baik pula proses kalibrasi yang dilakukan. Perbandingan juga dilakukan dengan membandingkan hasil prediksi pada gambar sebelum dikalibrasi dengan gambar yang sudah melalui proses kalibrasi, tujuannya adalah untuk melihat bagaimana

pengaruh kalibrasi ketika dilakukan prediksi menggunakan model P3NET. Hasil dari prediksi model P3NET adalah sebagai berikut.

dimana A = pigmen klorofil, B = pigmen karotenoid, dan C = pigmen antosianin.

Tabel 4.18 Hasil Prediksi Pigmen *Duranta Erecta* Kalibrasi KNN Set 1

Nama Dataset	KNN Set 1					
	Original			Kalibrasi		
	A	B	C	A	B	C
Rendah Dekat	1.37230	0.10247	0.04021	1.37065	0.10450	0.04828
Rendah Jauh	1.48510	0.10307	0.03368	1.38873	0.10147	0.05388
Sedang Dekat	1.32851	0.10480	0.04118	1.35568	0.10967	0.03836
Sedang Jauh	1.34352	0.10378	0.04019	1.32474	0.10696	0.04143
Tinggi Dekat	1.16516	0.09784	0.05240	1.34774	0.10873	-0.00011
Tinggi Jauh	1.18173	0.09538	0.07677	1.06788	0.10006	0.05987
Standart Deviasi	0.12116	0.00374	0.01561	0.12019	0.00391	0.02130

Pada tabel 4.18 nilai standart deviasi yang dihasilkan tidak berbeda jauh, bahkan terdapat hasil dari kalibrasi yang memberikan nilai standart deviasi lebih. Melalui hasil tersebut dapat dilihat bahwa kalibrasi KNN set 1 yang dilakukan hampir tidak merubah hasil prediksi pigmen bila dibandingkan dengan gambar yang tidak dikalibrasi.

Tabel 4.19 Hasil Prediksi Pigmen *Duranta Erecta* Kalibrasi KNN Set 2

Nama Dataset	KNN Set 2					
	Original			Kalibrasi		
	A	B	C	A	B	C
Rendah Dekat	1.37230	0.10247	0.04021	1.47831	0.09294	0.03336
Rendah Jauh	1.48510	0.10307	0.03368	1.37992	0.09328	0.03135
Sedang Dekat	1.32851	0.10480	0.04118	1.37588	0.11092	0.04605
Sedang Jauh	1.34352	0.10378	0.04019	1.34013	0.10761	0.03797
Tinggi Dekat	1.16516	0.09784	0.05240	1.31698	0.10951	0.02893
Tinggi Jauh	1.18173	0.09538	0.07677	0.77428	0.06335	0.06108
Standart Deviasi	0.12116	0.00374	0.01561	0.25267	0.01801	0.01205

Tabel 4.19 adalah hasil prediksi pigmen untuk kalibrasi KNN set 2. Bila dilihat dari hasilnya, nilai standart deviasi yang diberikan juga tidak mengalami perubahan yang sangat signifikan. Pada hasil kalibrasi klorofil dan antosianin menghasilkan nilai lebih dibandingkan dengan gambar tanpa kalibrasi.

Tabel 4.20 Hasil Prediksi Pigmen *Duranta Erecta* Kalibrasi DNN

Nama Dataset	Deep Learning			Kalibrasi		
	Original					
	A	B	C	A	B	C
Rendah Dekat	1.37230	0.10247	0.04021	1.47370	0.09835	0.05003
Rendah Jauh	1.48510	0.10307	0.03368	1.52385	0.09974	0.04515
Sedang Dekat	1.32851	0.10480	0.04118	1.41442	0.10801	0.04851
Sedang Jauh	1.34352	0.10378	0.04019	1.37589	0.10590	0.04853
Tinggi Dekat	1.16516	0.09784	0.05240	1.36162	0.10491	0.04919
Tinggi Jauh	1.18173	0.09538	0.07677	0.07163	0.42037	0.47500
Standart Deviasi	0.12116	0.00374	0.01561	0.55785	0.12946	0.17421

Tabel 4.20 adalah hasil prediksi pigmen dengan kalibrasi DNN. Pada tabel ini nilai standar deviasi yang dihasilkan oleh gambar original lebih kecil dibandingkan dengan hasil kalibrasi. Hal tersebut menunjukkan bahwa prediksi pigmen pada hasil kalibrasi lebih variatif dibandingkan tanpa kalibrasi.

Tabel 4.21 Hasil Prediksi Pigmen *Piper Betle* Kalibrasi KNN Set 1

Nama Dataset	KNN Set 1					
	Original			Kalibrasi		
	A	B	C	A	B	C
Rendah Dekat	0.80386	0.10020	0.10241	0.91062	0.10028	0.08901
Rendah Jauh	0.61030	0.11533	0.14673	0.71328	0.10334	0.12059
Sedang Dekat	0.61635	0.10867	0.10138	0.80389	0.10008	0.08826
Sedang Jauh	0.73032	0.10072	0.09335	0.89391	0.09975	0.08480
Tinggi Dekat	0.83183	0.09819	0.08122	1.04252	0.10111	0.05760
Tinggi Jauh	0.76475	0.10463	0.07335	0.99488	0.10265	0.04506
Standart Deviasi	0.09401	0.00644	0.02568	0.12107	0.00147	0.02660

Berdasarkan hasil dari tabel 4.21 nilai standar deviasi yang dihasilkan, nilai dari hasil kalibrasi memberikan nilai yang tidak jauh berbeda dengan gambar tanpa kalibrasi. Hanya nilai pigmen karotenoid saja yang lebih kecil dari hasil kalibrasi.

Tabel 4.22 Hasil Prediksi Pigmen *Piper Betle* Kalibrasi KNN Set 2

Nama Dataset	KNN Set 2					
	Original			Kalibrasi		
	A	B	C	A	B	C
Rendah Dekat	0.80386	0.10020	0.10241	0.58124	0.10836	0.08007
Rendah Jauh	0.61030	0.11533	0.14673	0.64129	0.10535	0.08062
Sedang Dekat	0.61635	0.10867	0.10138	0.68277	0.10508	0.08803
Sedang Jauh	0.73032	0.10072	0.09335	0.78684	0.10433	0.06866
Tinggi Dekat	0.83183	0.09819	0.08122	1.00367	0.10212	0.05494
Tinggi Jauh	0.76475	0.10463	0.07335	0.95044	0.10275	0.04350
Standart Deviasi	0.09401	0.00644	0.02568	0.17154	0.00222	0.01713

Pada tabel 4.22 Nilai standar deviasi yang diberikan juga tidak memiliki perubahan yang signifikan, pada kalibrasi KNN set 2 hanya pigmen klorofil saja yang menunjukkan nilai lebih besar dibandingkan dengan gambar tanpa kalibrasi.

Tabel 4.23 Hasil Prediksi Pigmen *Piper Betle* Kalibrasi DNN

Nama Dataset	Deep Learning					
	Original			Kalibrasi		
	A	B	C	A	B	C
Rendah Dekat	0.80386	0.10020	0.10241	0.85683	0.09857	0.08797
Rendah Jauh	0.61030	0.11533	0.14673	0.79386	0.10323	0.07796
Sedang Dekat	0.61635	0.10867	0.10138	0.58464	0.11510	0.10037
Sedang Jauh	0.73032	0.10072	0.09335	0.89227	0.10170	0.07628
Tinggi Dekat	0.83183	0.09819	0.08122	1.01208	0.10025	0.06253
Tinggi Jauh	0.76475	0.10463	0.07335	0.93712	0.10251	0.04769
Standart Deviasi	0.09401	0.00644	0.02568	0.14777	0.00589	0.01856

Selanjutnya pada tabel 4.23 nilai standar deviasi yang dihasilkan juga sama yaitu tidak memiliki perubahan yang terlalu signifikan. Pada hasil kalibrasi, nilai pigmen klorofil sajalah yang lebih besar dibandingkan dengan gambar tanpa kalibrasi.

Secara keseluruhan hasil prediksi pigmen pada masing-masing algoritma kalibrasi menunjukkan hasil yang tidak signifikan. Beberapa hasil kalibrasi bahkan memberikan nilai standar deviasi lebih besar dari gambar normal. Pada algoritma KNN hal tersebut banyak terjadi yang terutama pada pigmen klorofil dimana nilai standar deviasi dari gambar asal lebih kecil dibandingkan hasil kalibrasi.

## **BAB V**

### **KESIMPULAN DAN SARAN**

#### **5.1 Kesimpulan**

Proyek praktik kerja lapangan yang berjudul "Implementasi Kecerdasan Buatan Untuk Kalibrasi Citra Tanaman pada Proses Prediksi Kandungan Pigmen Fontosintesis Secara Non Destruktif dan In-Situ" ini sudah berjalan dengan cukup baik. Berdasarkan dari hasil pengujian yang dilakukan diperoleh simpulan sebagai berikut.

1. Pengembangan teknik kalibrasi secara "unsupervised" dapat dilakukan menggunakan metode *white-balance*. Metode tersebut sangat baik untuk menyamakan warna agar terlihat lebih natural tanpa harus melakukan segmentasi atau pengolahan *background* citra tanaman daun. Melalui penggunaan algoritma KNN dan DNN dapat diketahui seberapa baik metode *white-balance* diterapkan untuk tanaman daun.
2. Berdasarkan dari hasil uji *similarity* algoritma DNN memberikan hasil yang lebih baik bila dibandingkan dari algoritma KNN. Hal tersebut dapat dilihat pada hasil algoritma DNN yang nilai SSIM dan RMSEnya memberikan hasil yang lebih bagus dimana nilai SSIMnya semakin mendekati nilai satu dan RMSEnya juga semakin mengecil. Sedangkan untuk hasil prediksi pigmen, kedua algoritma memberikan hasil standar deviasi yang tidak terlalu berbeda jauh dengan gambar asal. Secara visual gambar yang dihasilkan dari algoritma DNN memberikan hasil yang lebih natural dibandingkan dengan algoritma KNN.

#### **5.2 Saran**

Berdasarkan kesimpulan yang sudah diperoleh, maka terdapat beberapa saran yang bisa dicoba untuk memperbaiki teknik kalibrasi pada citra gambar daun ini yaitu sebagai berikut.

1. Sebelum dilakukan kalibrasi ada baiknya memilih parameter yang sesuai agar gambar yang dilakukan kalibrasi sesuai dengan harapan dan tidak terjadi *over-kalibrasi*. Parameter yang dapat dirubah adalah seperti jenis dataset dan teknik *gamut mapping* yang digunakan khususnya pada algoritma KNN. Sedangkan untuk algoritma DNN parameter yang dapat dirubah adalah model arsitektur itu sendiri, perlu dilakukan beberapa pengujian *training* untuk mendapatkan hasil yang diharapkan untuk pengolahan kalibrasi citra daun
2. Pengolahan dataset bisa saja memberikan hasil kalibrasi yang lebih baik apabila dataset *training* yang digunakan adalah citra gambar daun itu sendiri. Cara pengumpulan dataset itu sendiri adalah dengan cara menfoto dataset daun bersertakan *color chart* untuk mendapatkan gambar *ground-truthnya*. Setelah gambar diambil maka tahap selanjutnya adalah mengambil RGB-uvnya yang kemudian disederhanakan dengan menggunakan teknik PCA.

## DAFTAR PUSTAKA

Afifi, M. & Brown, M. S., 2020. *Deep White Balance*. [Online] Available at: [https://github.com/mahmoudnafifi/Deep\\_White\\_Balance](https://github.com/mahmoudnafifi/Deep_White_Balance)

Afifi, M. & Michael Brown, S., 2020. Deep White-Balance Editing. *Computer Vision Foundation*, pp. 1397-1406.

Afifi, M. & Price, B., 2019. *WB sRGB*. [Online] Available at: [https://github.com/mahmoudnafifi/WB\\_sRGB](https://github.com/mahmoudnafifi/WB_sRGB)

Bevans, M. L. & Chattman, B., 2004. Method For Color Correction of Digital Images. *International Application*.

Bianco, G. & Muzzupappa, M., 2015. A New Color Correction Method for Underwater Imaging. *Computer Vision and Robotics Conference Paper*.

Daniel, J., 2021. *Back Propagation Neural Network: What is Backpropagation Algorithm in Machine Learning?*. [Online] Available at: <https://www.guru99.com/backpropogation-neural-network.html>

Gasparini, F. & Schettini, R., 2014. Unsupervised Color Correction for Digital Photographs. *Color Correction for Digital Photographs*, pp. 1-6.

Jaadi, Z., 2021. *A Step-by-Step Explanation of Principal Component Analysis (PCA)*. [Online] Available at: <https://builtin.com/data-science/step-step-explanation-principal-component-analysis>

Johnson, J., 2020. *What's a Deep Neural Network? Deep Nets Explained*. [Online] Available at: <https://www.bmc.com/blogs/deep-neural-network/>

Kirshi, Y., 2021. *How To Measure Image Similarities in Python*. [Online] Available at: <https://betterprogramming.pub/how-to-measure-image-similarities-in-python-12f1cb2b7281>

Mahmoud, A., Brian, P., Scott, C. & Michael, S. B., 2019. When Color Constancy Goes Wrong: Correcting Improperly White-Balanced Images. pp. 1-10.

Mahmoud, A. & Michael, S. B., 2020. Deep White-Balance Editing. *Computer Vision Foundation*, pp. 1397-1406.

Młodzińska, E., 2009. Acta Biologica Cracoviensia Series Botanica. *Survey of Plant Pigments: Molecular and Environmental*, pp. 7-16.

Neupane, J., 2019. Agronomic Basis and Strategies for Precision Water Management: A Review. *Agronomy* 9, pp. 1-21.

Peterson, L. E., 2009. *K-nearest neighbor*. [Online] Available at: [http://scholarpedia.org/article/K-nearest\\_neighbor#Characteristics\\_of\\_kNN](http://scholarpedia.org/article/K-nearest_neighbor#Characteristics_of_kNN)

Pragati, B., 2021. *The Essential Guide to Neural Network Architectures*. [Online] Available at: <https://www.v7labs.com/blog/neural-network-architectures-guide>

rys, 2018. *Cara Kerja Pencahayaan Dan White Balance Pada Kamera*. [Online] Available at: <https://rekreative.com/cara-kerja-pencahayaan-dan-white-balance-pada-kamera/>

Samuel Eichhorn, B. & Wu, B.-S., 2019. An Update on Plant Photobiology and Implications for Cannabis Production. *frontiers in Plant Science*, pp. 1-3.

Seb, 2021. *Principal Components Analysis Explained for Dummies*. [Online] Available at: <https://programmatically.com/principal-components-analysis-explained-for-dummies/>

Shafi, U. & Mumtaz, R., 2019. Precision Agriculture: Techniques and Practices. pp. 1-25.

Suliztia, M. L., 2018. *Comparing Naive Bayes and K-Nearest Neighbor Classification Methods of Breast Cancer in Coimbra, Portugal - 2018 with Python*. [Online] Available at: <https://medium.com/@megaluna/comparing-naive-bayes-and-k-nearest-neighbor-classification-methods-of-breast-cancer-in-coimbra-portugal-2018-with-python-5a2a2a2a2a2a>

k-nearest-neighbor-classification-methods-of-breast-cancer-in-coimbra-  
bba254f04860

Tarragó-Celada, J. & M., J., 2019. *Colour, chlorophyll and chromatography*.

[Online]

Available at: <https://www.scienceinschool.org/article/2019/colour-chlorophyll-and-chromatography/>