**ANALISIS PENGARUH RUANG WARNA TERHADAP AKURASI PENGUKURAN KADAR PIGMEN FOTOSINTESIS DENGAN MENGGUNAKAN CNN TERHADAP PERMASALAHAN *COLOR CONSTANCY***

**TUGAS AKHIR**



**ADN AGUNG ROCHMAN ARIFIN**

**311510002**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS SAINS DAN TEKNOLOGI**

**UNIVERSITAS MA CHUNG**

**MALANG**

**2019**

# DAFTAR ISI

DAFTAR ISI i

DAFTAR GAMBAR iv

DAFTAR TABEL vii

DAFTAR PERSAMAAN viii

Bab I Pendahuluan 1

1. Latar Belakang 1
2. Identifikasi Masalah 3
3. Batasan Masalah 3
4. Rumusan Masalah 3
5. Tujuan 4
6. Manfaat 4
7. Rencana Penelitian 4

Bab II Tinjauan Pustaka 6

1. Fotosintesis 6
2. Pigmen Fotosintesis 7
3. Metode Kuantifikasi Pigmen Fotosintesis 9
4. Model Ruang Warna 12
5. Color Constancy 17
6. Metode Inpainting Navier Stokes 18
7. Deep Learning 18
8. CNN 19
9. Python 25
10. Flask 26
11. TensorFlow 26
12. Keras 27
13. OpenCv 29
14. Numpy 29
15. Flat File 30
16. Penelitian Terdahulu 30

Bab III Analisis dan Perancangan Sistem 33

1. Analisis Masalah 34
2. Pengumpulan Data 34
3. Pembuatan Dataset 36
4. Desain Arsitektur CNN 37
5. Perancangan Desain Sistem 38

3.5.1 Flowchart Alur Kerja Sistem 39

3.5.2 Use Case Diagram Sistem 46

3.5.3 Sequence Diagram Sistem 46

3.5.4 Desain Flat File Database 52

1. Implementasi Program 56
2. Desain Antarmuka 56
3. Pengujian Akurasi CNN P3Net Terhadap Ruang Warna 60
4. Pengujian Antarmuka Aplikasi Web 62

Bab IV Hasil dan Pembahasan 64

1. Ruang Warna Tunggal Dengan Akurasi Terbaik Tanpa Proses Inpaint 64
2. Ruang Warna Kombinasi Dengan Akurasi Terbaik Tanpa Proses

Inpaint 65

1. Perbandingan Akurasi Ruang Warna Tunggal dan Kombinasi Tanpa Proses Inpaint 66
2. Ruang Warna Tunggal Dengan Akurasi Terbaik Dengan Proses

Inpaint 67

1. Ruang Warna Kombinasi Dengan Akurasi Terbaik Dengan Proses

Inpaint 68

1. Perbandingan Akurasi Ruang Warna Tunggal dan Kombinasi Dengan Proses Inpaint 69
2. Perbandingan Antara Ruang Warna Terbaik Yang Melalui Proses

Inpaint dan Tidak Melalui Proses Inpaint 71

1. Tampilan Antarmuka Aplikasi 72
2. Hasil Pengujian Antarmuka Aplikasi Web 77

Bab V Simpulan dan Saran 82

1. Simpulan 82
2. Saran 83

DAFTAR PUSTAKA 84

Lampiran 1 86

# DAFTAR GAMBAR

Gambar 1.1 Selisih komponen ruang warna RGB (b), HSV (c), LAB (d), dan

YCbCr (e) dari citra original dengan pencahayaan berbeda (a) 2

Gambar 2.1 Proses reaksi terang dan reaksi gelap 7

Gambar 2.2 Struktur kimia pigmen-pigmen klorofil dan karotenoid pada

membran thylakoid 8

Gambar 2.3 Absorbsi spektra pigmen pada tanaman 8

Gambar 2.4 Spektrofotometer Uv Vis 10

Gambar 2.5 Spektrometer optik 11

Gambar 2.6 Kubus ruang warna RGB 13

Gambar 2.7 Segitiga ruang warna HSI 13

Gambar 2.8 Hexagon ruang warna HSV 14

Gambar 2.9 Ruang warna LAB 15

Gambar 2.10 Kubus RGB di dalam ruang warna YCbCr 16

Gambar 2.11 Objek di bawah *sunlight* (a) dan di bawah *skylight* 17

Gambar 2.12 Matriks citra (kiri) dan matriks filter (kanan) 19

Gambar 2.13 Hasil *feature map* dari lapisan konvolusi 20

Gambar 2.14 Gambar citra asli (kiri), gambar citra hasil konvolusi (tengah), dan gambar citra hasil lapisan non-linearitas (kanan) 20

Gambar 2.15 Proses *pooling* menggunakan *max pooling* 21

Gambar 2.16 *Array* sebelum diproses pada lapisan pemipihan (kiri) dan array

setelah mengalami proses pemipihan (kanan) 21

Gambar 2.17 Arsitektur ShallowNet 31

Gambar 2.18 Arsitektur LeNet 31

Gambar 2.19 Arsitektur AlexNet 31 Gambar 3.1 Alur penelitian 33 Gambar 3.2 Data citra sampel daun 35 Gambar 3.3 Arsitektur model CNN 38

Gambar 3.4 Flowchart alur kerja fungsi sign up 39

Gambar 3.5 Flowchart alur kerja fungsi delete user 40

Gambar 3.6 Flowchart alur kerja fungsi sign in 41

Gambar 3.7 Flowchart alur kerja fungsi latih model 42

Gambar 3.8 Flowchart alur kerja fungsi estimasi pigmen 43

Gambar 3.9 Flowchart alur kerja fungsi estimasi pigmen (lanjutan) 44

Gambar 3.10 Flowchart alur kerja fungsi lihat dan hapus data record 45

Gambar 3.11 Use case diagram sistem estimasi kadar pigmen fotosintesis 46

Gambar 3.12 Sequence diagram fungsi sign in 47

Gambar 3.13 Sequence diagram fungsi delete user 47

Gambar 3.14 Sequence diagram fungsi sign up 48

Gambar 3.15 Sequence diagram fungsi latih model 49

Gambar 3.16 Sequence diagram fungsi estimasi kadar pigmen 50

Gambar 3.17 Sequence diagram fungsi lihat dan hapus record data latih 51

Gambar 3.18 Halaman “Latih Model” bagian pembuatan project atau

buka project 56

Gambar 3.19 Halaman “Latih Model” bagian proses latih model 57

Gambar 3.20 Halaman “Latih Model” bagian hasil latih model 57

Gambar 3.21 Halaman “Estimasi” bagian pengunggahan citra daun dan hasil

estimasi 58

Gambar 3.22 Halaman “Estimasi” bagian rubah model estimasi ruang warna 58

Gambar 3.23 Halaman “Estimasi” bagian pilihan project yang menyediakan

model 59

Gambar 3.24 Halaman “Estimasi” bagian pilihan model 59

Gambar 3.25 Halaman “Record” bagian pilihan project 60

Gambar 3.26 Halaman “Record” bagian rekap data 60

Gambar 4.1 Grafik urutan nilai akurasi latih model ruang warna tunggal tanpa

inpaint 64

Gambar 4.2 Grafik urutan nilai akurasi latih model ruang warna kombinasi

tanpa inpaint 65

Gambar 4.3 Grafik urutan nilai akurasi latih model ruang warna tunggal dan

kombinasi tanpa inpaint 66

Gambar 4.4 Grafik urutan nilai akurasi latih model ruang warna tunggal dan

kombinasi tanpa inpaint (lanjutan) 67

Gambar 4.5 Grafik urutan nilai akurasi latih model ruang warna tunggal

dengan inpaint 68

Gambar 4.6 Grafik urutan nilai akurasi latih model ruang warna kombinasi

dengan inpaint 69

Gambar 4.7 Grafik urutan nilai akurasi latih model ruang warna tunggal dan

kombinasi dengan inpaint 70

Gambar 4.8 Grafik urutan nilai akurasi latih model ruang warna tunggal dan

kombinasi dengan inpaint (lanjutan) 70

Gambar 4.9 Hasil tampilan halaman “Home” 72

Gambar 4.10 Hasil tampilan halaman “Latih Model” tahap awal 73

Gambar 4.11 Hasil tampilan halaman “Latih Model” bagian create project 74

Gambar 4.12 Hasil tampilan halaman “Latih Model” bagian open new project 74

Gambar 4.13 Hasil tampilan halaman “Latih Model” bagian latih model 74 Gambar 4.14 Hasil tampilan halaman “Estimasi” bagian estimasi 75

Gambar 4.15 Hasil tampilan halaman “Estimasi” bagian estimasi (lanjutan) 75

Gambar 4.16 Hasil tampilan halaman “Record” bagian pilih project 76

Gambar 4.17 Hasil tampilan halaman “Record” bagian rekap data 77

Gambar 4.18 Hasil tampilan halaman “Record” bagian rekap data (lanjutan) 77

Gambar 4.19 Grafik hasil penilaian tema warna web 78

Gambar 4.20 Grafik hasil penilaian tampilan dan pengaturan komponen pada

web 78

Gambar 4.21 Grafik hasil penilaian kejelasan informasi pada website 79 Gambar 4.22 Grafik hasil penilaian alur penggunaan aplikasi web 79

Gambar 4.23 Grafik hasil penilaian kemudahan proses input 80

Gambar 4.24 Grafik hasil penilaian fungsionalitas aplikasi web 80

Gambar 4.25 Grafik hasil penilaian kejelasan informasi pada website 80

# DAFTAR TABEL

Tabel 1.1 Rencana penelitian 4

Tabel 3.1 Data kadar pigmen fotosintesis 36

Tabel 3.2 Data kadar pigmen ternormalisasi 36

Tabel 3.3 Arsitektur model CNN 38

Tabel 3.4 Keterangan field flat file model\_select 53 Tabel 3.5 Contoh data flat file model\_select 53 Tabel 3.6 Keterangan field flat file project 53 Tabel 3.7 Contoh data flat file project 53

Tabel 3.8 Keterangan field flat file user 54 Tabel 3.9 Contoh data flat file user 54

Tabel 3.10 Keterangan field flat file ruang warna 55 Tabel 3.11 Contoh data flat file ruang warna 55

Tabel 3.12 Aspek pengujian model CNN 61

Tabel 3.13 Aspek pengujian antarmuka aplikasi web 62

Tabel 4.1 Urutan nilai akurasi latih model ruang warna tunggal tanpa

inpaint 64

Tabel 4.2 Urutan nilai akurasi latih model ruang warna kombinasi tanpa

inpaint 66

Tabel 4.3 Urutan nilai akurasi latih model ruang warna tunggal dan kombinasi

tanpa inpaint 67

Tabel 4.4 Urutan nilai akurasi latih model ruang warna tunggal dengan

inpaint 68

Tabel 4.5 Urutan nilai akurasi latih model ruang warna kombinasi dengan

inpaint 69

Tabel 4.6 Urutan nilai akurasi latih model ruang warna tunggal dan kombinasi

dengan inpaint 71

Tabel 4.7 Urutan nilai akurasi terbaik semua latih model ruang warna dan

Preprocessing inpaint 72

# DAFTAR PERSAMAAN

Persamaan 2.1 Fotosintesis reaksi terang 6

Persamaan 2.2 Fotosintesis reaksi gelap 6

Persamaan 2.3 Reaksi fotosintesis secara keseluruhan 6

Persamaan 2.4 Energi cahaya 8

Persamaan 2.5 Panjang gelombang cahaya 9

Persamaan 2.6 Energi dari panjang gelombang cahaya 9

Persamaan 2.7 *Intensity* HSI 14

Persamaan 2.8 *Hue* HSI 14

Persamaan 2.9 Nilai pada formula hue HSI 14

Persamaan 2.10 *Saturation* HSI 14

Persamaan 2.11 *Value* HSV 15

Persamaan 2.12 *Saturation* HSV 15

Persamaan 2.13 *Hue* HSV 15

Persamaan 2.14 Konversi RGB ke YCbCr 16

Persamaan 2.15 Perbaruan parameter dengan SGD 22

Persamaan 2.16 Perbaruan parameter dengan Adagrad 23

Persamaan 2.17 RMS gradien 23

Persamaan 2.18 Parameter delta Adadelta 23

Persamaan 2.19 Perbaruan parameter Adadelta 23

Persamaan 2.20 *Running average* gradien kuadrat RMSProp 23

Persamaan 2.21 Perbaruan parameter RMSProp 23

Persamaan 2.22 Momen selanjutnya dari gradien Adam 23

Persamaan 2.23 *Bias corrected past average gradient* Adam 24

Persamaan 2.24 Momen awal dari gradien Adam 24

Persamaan 2.25 *Bias corrected past average epoch* Adam 24

Persamaan 2.26 Perbaruan parameter Adam 24

Persamaan 2.27 Perbaruan parameter Adamax 24

Persamaan 2.28 Norma tidak terbatas Adamax 24

Persamaan 2.29 Perbaruan parameter Nadam 24

Persamaan 2.30 *Mean squared error* 24

Persamaan 2.31 *Mean absolute error* 25

Persamaan 2.32 *Mean absolute percentage error* 25

Persamaan 2.33 *Mean squared logarithmic error* 25

Persamaan 2.34 *Chlorophyll reflectance index - Green* 31

Persamaan 2.35 *Chlorophyll reflectance index – Red Edge* 31

Persamaan 2.36 *Carotenoid reflectance index - Green* 31

Persamaan 2.37 *Carotenoid reflectance index – Red Edge* 31

Persamaan 2.38 *Anthocyanin reflectance index* 31

Persamaan 2.39 *MSE -* Prilianti 32

# Bab 1

**Pendahuluan**

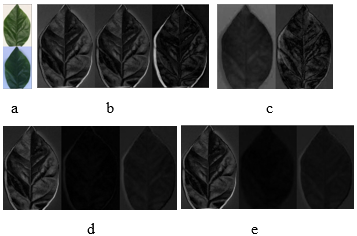
## 1.1 Latar Belakang

Metabolisme tanaman dilakukan dalam proses kimia yang bernama fotosintesis. Cara kerja fotosintesis yaitu dengan memunculkan reaksi kimia menggunakan energi cahaya yang mengolah karbon dioksida dan air kemudian menghasilkan glukosa dan zat buangan berupa oksigen. Glukosa yang dihasilkan kemudian dapat digunakan sebagai energi untuk metabolisme tanaman atau dirubah dalam bentuk pati yang disimpan dan digunakan pada saat membutuhkan (Baker, 2007). Dalam proses fotosintesis tanaman terdapat molekul-molekul pigmen yang bertanggung jawab terhadap penyerapan energi cahaya untuk reaksi kimia yaitu, klorofil dan karotenoid. Pada dasarnya molekul pigmen menangkap dan memantulkan energi cahaya dengan spektrum tertentu. Energi cahaya yang diserap digunakan untuk mendukung proses fotosintesis sedangkan energi cahaya yang dipantulkan menimbulkan efek warna pada tanaman yang terbatas pada panjang gelombang energi cahaya yang dipantulkan. Pigmen klorofil merupakan pigmen hijau yaitu pigmen yang memantulkan spektrum warna hijau. Untuk pigmen karotenoid, spektrum cahaya yang dipantulkan meliputi spektrum warna merah, oranye, dan kuning (Speer, 1997).

Pengukuran kadar pigmen pada tumbuhan pada umumnya dapat dilakukan dengan ekstraksi laboraturium menggunakan analisa spektrofotometrik yang mengukur nilai absorbansi dan reflektansi larutan pigmen daun, atau menggunakan HPLC (*High Performance Liquid Chromatography*). Kedua metode tersebut memberikan hasil yang akurat, perbedaannya terletak pada hasil pengukuran. Untuk analisa spektrofotometrik hanya dapat memisahkan hasil pigmen klorofil a dan klorofil b, namun tidak untuk pigmen lain seperti karotenoid atau antosianin. Sedangkan untuk HPLC, analisa yang dihasilkan dapat memberikan informasi masing-masing pigmen. Kekurangan dari kedua metode pengukuran tersebutadalah waktu yang dibutuhkan relatif cukup lama, memerlukan biaya mahal, dan bersifat

desktruktif yaitu daun harus dihancurkan untuk diekstrak pigmennya (Croft, 2017). Maka dari itu diperlukan suatu metode pengukuran kadar pigmen tanaman yang mudah, cepat, murah, dan akurat.

Metode yang ditawarkan adalah pengukuran pigmen menggunakan citra digital yang diambil menggunakan kamera *smartphone* yang kemudian diproses menggunakan metode *deep learning* yang menghasilkan estimasi kadar pigmen dari daun. Sudah ada penelitian sebelumnya (Prilianti dkk, 2018) yang membahas tentang pengukuran kadar pigmen (klorofil, karotenoid, antosianin) pada tanaman menggunakan citra digital dan *deep learning*, namun penelitian hanya sebatas 1 jenis ruang warna yaitu RGB (*Red, Green, Blue*). Maka dari itu, penelitian ini merancang sistem pengukuran kadar pigmen tanaman dengan citra digital dan *deep learning* dengan menggunakan kombinasi beberapa ruang warna berbeda yaitu RGB, HSV (*Hue, Saturation, Value*), LAB, dan YCbCr sebagai input.



Gambar 1. 1 Selisih komponen ruang warna RGB (b), HSV (c), LAB(d), dan YCbCr (e) dari citra original dengan pencahayaan berbeda (a)

Implementasi ruang warna yang berbeda mempertimbangkan pengaruh fenomena *color constancy* yaitu objek dibawah iluminasi yang berbeda terhadap akurasi pengukuran. Pertimbangan pemilihan ruang warna didasarkan oleh pengaruh pencahayaan terhadap objek dengan iluminasi berbeda. Hal ini ditunjukan dengan Gambar 1.1 yang memperlihatkan tingkat kesamaan pada penggunaan model warna LAB, HSV, dan YCbCr. Selain itu, terdapat warna putih pada citra daun yang merupakan refleksi cahaya permukaan daun. Perbedaan ruang warna dan keberadaan refleksi cahaya putih pada citra daun menjadi perbandingan yang dianalisa untuk mengetahui pengaruhnya terhadap hasil akurasi dari CNN sehingga dapat didapatkan informasi ruang warna yang paling baik terhadap fenomena *color constancy* dan refleksi cahaya putih oleh permukaan daun untuk perancangan sistem pengukuran kadar pigmen menggunakan citra digital dan *deep learning* yang baik.

## 1.2 Identifikasi Masalah

Berdasarkan latar belakang di atas, masalah yang dapat diidentifikasi adalah perlunya analisa pengaruh ruang warna berbeda terhadap fenomena *color constancy* kepada akurasi pengukuran kadar pigmen menggunakan CNN dengan masukan citra digital daun.

## 1.3 Batasan Masalah

Adapun batasan masalah dalam tugas akhir sebagai berikut:

1. Data citra diambil dengan pencahayaan ruangan atau pencahayaan yang cukup.
2. Ruang warna yang digunakan dalam penelitian adalah RGB, HSV, LAB, dan YCbCr.
3. Penanganan refleksi cahaya berwarna putih pada citra daun dilakukan dengan menerapkan algoritma metode *inpaint* Navier Stokes.
4. Menggunakan bahasa pemrograman Python, *back-end* TensorFlow dan API Keras.
5. *Platform* antarmuka aplikasi sebagai media eksperimen adalah *website*.

## 1.4 Rumusan Masalah

Bagaimana cara untuk melakukan analisis pengaruh ruang warna terhadap akurasi pengukuran kadar pigmen fotosintesis menggunakan CNN dengan mempertimbangkan masalah *color constancy*.

## 1.5 Tujuan

Tujuan yang dicapai dalam tugas akhir ini adalah melakukan analisis pengaruh ruang warna terhadap akurasi pengukuran kadar pigmen fotosintesis menggunakan CNN dengan mempertimbangkan masalah *color constancy* menggunakan bahasa pemrograman Python.

## 1.6 Manfaat

Adapun manfaat yang didapatkan dari tugas akhir ini adalah sebagai berikut:

1. Bagi pengembang, dapat menjadi referensi dan pertimbangan penggunaan ruang warna untuk mengatasi masalah *color constancy* dalam penyelesaian kasus serupa menggunakan pendekatan metode CNN.
2. Bagi pembaca, dapat menjadi referensi sebagai bahan ajar dan menjadi kajian untuk penelitian selanjutnya.
3. Bagi penulis, sebagai kesempatan untuk menambah wawasan dalam pemahaman pengaruh ruang warna untuk mengatasi masalah *color constancy* terhadap akurasi pengukuran kadar pigmen menggunakan citra digital dan CNN.

## 1.7 Sistematika Penulisan

Sistematika pada laporan Tugas Akhir dibagi dalam lima bab sebagai berikut:

Bab I : Pendahuluan

Bab ini berisi tentang latar belakang adanya penelitian, identifikasi masalah, batasan masalah, rumusan masalah, tujuan, luaran, manfaat dan sistematika penulisan.

Bab II : Tinjauan Pustaka

Bab ini berisi tentang latar belakang adanya penelitian, identifikasi masalah, batasan masalah, rumusan masalah, tujuan, luaran, manfaat dan sistematika penulisan.

Bab III : Analisis dan Perancangan Sistem

Bab ini berisi tentang latar belakang adanya penelitian, identifikasi masalah, batasan masalah, rumusan masalah, tujuan, luaran, manfaat dan sistematika penulisan.

Bab IV : Metodelogi Penelitian

Bab ini berisi tentang latar belakang adanya penelitian, identifikasi masalah, batasan masalah, rumusan masalah, tujuan, luaran, manfaat dan sistematika penulisan.

Bab V : Hasil dan Saran

Bab ini berisi tentang latar belakang adanya penelitian, identifikasi masalah, batasan masalah, rumusan masalah, tujuan, luaran, manfaat dan sistematika penulisan.

# Bab II

**Tinjuan Pustaka**

## 2.1 Fotosintesis

Menurut Johnson (2016), fotosintesis merupakan proses kimia pada banyak organisme yang kemudian menghasilkan oksigen dan makanan yang mendukung keberlanjutan kehidupan pada planet bumi. Oksigen yang dihasilkan dari proses fotosintesis memungkinkan terjadinya pembentukan lapisan ozon, evolusi respirasi aerobik, dan kehidupan multiseluler kompleks. Proses fotosintesis melibatkan pengubahan air dan menjadi molekul kompleks yaitu dan karbohidrat. Proses fotosintesis dapat dibagi menjadi 2 yaitu reaksi terang dan reaksi gelap. Fotosintesis reaksi terang menggunakan cahaya dalam proses pemisahan air menjadi oksigen, proton, dan elektron, sedangkan pada fotosintesis reaksi gelap, proton dan elektron digunakan untuk mengubah menjadi karbohidrat. Kedua proses ini ditunjukan dengan persaman 2.1, 2.2, dan secara keseluruhan ditunjukan dengan persamaan 2.3.

1. Fotosintesis reaksi terang

(2.1)

1. Fotosintesis reaksi gelap

(2.2)

1. Reaksi fotosintesis secara keseluruhan

(2.3)

Pada tanaman daun adalah organ fotosintesis yang merupakan bagian terluas yang menyediakan sel hijau untuk masuknya dan cahaya yang dikendalikan dengan lubang-lubang kecil pada lapisan epidermis yang lebih rendah yang disebut stomata. Stomata dapat terbuka dan tertutup menyesuakian dengan kandungan air pada tanaman, pada saat terhidrasi stomata terbuka agar masuk dan tertutup ketika kandungan air pada tanaman sedikit untuk mencegah berkurang

kandungan air pada tanaman. Terdapat sel kloroplas dalam sel hijau pada daun yang merupakan tempat terjadinya fotosintesis. Pada membran ketiga kloroplas yang disebut membran thylakoid terjadi reaksi cahaya yang melibatkan transfer elektron dan proton. Sedangkan reaksi gelap terjadi pada stroma melalu siklus Calvin-Benson. Reaksi cahaya merupakan transfer elektron dari air menuju untuk membentuk NADPH yang dikombinasikan dengan transfer proton sehingga terjadi fosforilasi dari *adenosine diphosphate* (ADP) menjadi *adenosine triphosphate* (ATP). Kemudian ATP dan NADPH digunakan pada siklus Calvin-Benson untuk mengubah menjadi karbohidrat, yang menghasilkan lagi ADP dan . Dapat disimpulkan bahwa proses fotosintesis sangat bergantung pada penyerapan cahaya yang dilakukan dengan molekul-molekul pigmen pada membran thylakoid yaitu klorofil, karotenoid, dan antocyanin. Proses fotosintesis pada reaksi terang dan gelap diilustrasikan dengan gambar 2.1.



Gambar 2.1 Proses reaksi terang dan reaksi gelap (Johnson, 2016)

## 2.2 Pigmen Fotosintesis

Menurut Johnson (2016), fotosintensis sangat bergantung dengan mekanisme fotosistem pada tanaman, yaitu komponen-komponen yang bertanggung jawab pada proses penyerapan cahaya. Salah satu komponen fotosistem adalah pigmen penyerap cahaya pada tanaman. Pigmen pada tanaman adalah klorofil, karotenoid, dan antosianin. Struktur kimia dari pigmen klorofil dan karotenoid yang memungkinkan proses penyerapan cahaya ditunjukan pada Gambar 2.2.



Gambar 2. 2 Struktur kimia pigmen-pigmen klorofil dan karotenoid pada membran thylakoid (Johnson, 2016)

Variasi pigmen pada tanaman di daratan yaitu klorofil *a* dan *b*, karetonoid, dan antosianin. Klorofil menyerap cahaya biru dan merah sehingga terlihat berwarna hijau, sedangkan karotenoid hanya menyerap cahaya biru sehingga terlihat berwarna kuning atau merah. Spektrum penyerapan pigmen ditunjukan pada Gambar 2.3.



Gambar 2. 3 Absorbsi spektra pigmen pada tanaman (Féret dkk., 2017)

Cahaya atau radiasi elektromagnetik mempunyai sifat sebagai partikel (kuanta cahaya) dan gelombang. Nilai tiap energi pada kuanta cahaya dengan jumlah diskrit dapat didapatkan dengan mengkalikan konstanta Planck, *h* (6.626×10 − 34 J·s) dengan *v* yaitu frekuensi radiasi dalam siklus per detik (s − 1) pada persamaan:

*E* = *hv* (2.4)

Frekuensi (*v*) cahaya dan energinya berbeda sesuai dengan waranya, sehingga foton biru (∼450 nm) lebih berenergi dibandingkan dengan foton merah (∼650 nm). Hubungan antar frekuensi dan panjang gelombang (λ) dijelaskan pada persamaan berikut dengan *c* sebagai kecepatan cahaya (3.0×108 m·s − 1):

*λ = c/ν* (2.5)

Energi dari panjang gelombang cahaya tertentu didapatkan dengan:

*E = hc/λ* (2.6)

Sehingga foton merah (∼650 nm) memiliki energi 176 kJ·mol − 1.

Elektron-elektron dalam sistem π dari pigmen dapat melompat (tereksitasi) dari orbit molekul rendah ke orbit molekul yang lebih tinggi melalui penyerapan panjang gelombang cahaya tertentu dari spektrum yang dapat dilihat (400-725 nm). Terdapat dua kondisi tereksitasi S1 dan S2 pada klorofil. Pada proses eksitasi dari S0 ke S1 energi yang digunakan berasal dari foton merah (600-700nm), sedangkan untuk perubahan dari S0 ke S2 membutuhkan energi yang lebih yaitu dari foton biru (400-500nm) dengan panjang gelombang lebih pendek.

## 2.3 Metode Kuantifikasi Pigmen Fotosintesis

Dalam menghitung nilai kadar pigmen fotosintesis terdapat berbagai macam metode yang dapat digunakan, yaitu:

1. Desktruktif

Menurut Hasibuan (2015), spektrofotometer merupakan alat untuk mengukur nilai absorbansi suatu sampel sebagai fungsi panjang gelombang, dengan memanfaatkan sifat media yang menyerap cahaya pada panjang gelombang tertentu. Prinsip kerja spektrofotometer adalah bila cahaya jatuh pada suatu medium homogen, sebagain dari sinar cahaya akan dipantulkan dan diserap dalam medium, dan sisanya diteruskan. Cahaya yang diteruskan kemudian dinyatakan dalam nilai absorbansi karena memiliki hubungan dengan konsentrasi sampel.



Gambar 2. 4 Spektrofotometer Uv Vis

Contoh alat spektrofotometer ditunjukan dengan gambar 2.4. Berikut adalah bagian-bagian dari spektrofotometer.

1. Sumber cahaya

Sumber cahaya yang dipakai pada spektrofotometer harus memiliki pancaran radiasi yang stabil dan intensitasnya tinggi. Sumber cahaya didapat dari lampu pijar dengan kawat rambut terbuat dari tungsten yang menghasilkan cahaya dengan panjang gelombang 350-2200 nm.

1. Monokromator

Monokromator adalah alat untuk menggerakkan cahaya polikromatis menjadi beberapa komponen panjang gelombang tertentu yang berbeda dengan melakukan dispersi.

1. Cuvet

Cuvet pada spektrofotometer berfungsi sebagai tempat sampel yang akan dianalisis dan sudah dihancurkan terlebih dahulu. Cuvet dapat terbuat dari kwarsa, plexuglass, kaca dan plastik dengan bentuk tabung empat persegi panjang.

1. Detektor

Detektor berfungsi untuk mengubah cahaya menjadi sinyal listrik yang selanjutnya ditampilkan datanya dalam bentuk angka sigital atau jarum penunjuk. Pengukuran transmitan larutan sampel menjadi cara untuk menentukan konsentrasinya dengan menggunakan hukum Lambert Beer.

1. Non-Destruktif
2. Spektrometer optik



Gambar 2. 5 Spektrometer optik

Spektrometer optik merupakan alat untuk mengukur nilai intensitas reflektansi cahaya tampak dari suatu benda. Batas atas panjang gelombang yang digunakan yaitu 850 nm, sedangkan untuk batas bawah 180 nm (Potts J. P., 1987). Gambar 2.5 menunjukan alat spektometer optik. Implementasi penggunaan spektrofotmeter optik dalam pengukuran kadar klorofil adalah dengan menghitung indeks reflektansi masing-masing pigmen. Nilai variabel yang dibutuhkan dalam perhitungan indeks reflektansi pigmen adalah intensitas cahaya pada gelombang tertentu yang dipantulkan oleh daun tanaman. (Prilianti dkk., 2018)

1. Citra digital

Analisa citra digital dapat digunakan sebagai metode pengukuran kadar pigmen fotosintesis. Implementasi metode ini dilakukan dengan mendapatkan data nilai warna dari citra daun tanaman yang kemudian dapat dianalisa korelasi antara data warna dengan kadar klorofil asli untuk mendapatkan model persamaan yang digunakan untuk mengestimasi kadar klorofil suatu citra daun (Borres dkk., 2017). Cara lain untuk mengukur kadar klorofil menggunakan data citra adalah dengan mengimplementasikan *machine learning*, seperti penelitian yang telah dilakukan oleh Prilianti, dkk (2018). Dalam penelitian tersebut, arsitektur CNN dengan data RGB dari citra daun tanaman digunakan untuk memprediksi nilai kadar pigmen fotosintesis dari daun.

## 2.4 Model Ruang Warna

Menurut Ahmad (2005), model ruang warna merupakan model untuk mendefinisikan dan menampilkan warna-warna pada berbagai macam media. Secara umum terdapat dua macam model ruang warna yang digunakan dalam dunia komputer grafik yaitu model RGB yang digunakan pada monitor dan model CMY(K) yang digunakan pada kebanyakan mesin pencetak dokumen (*printer*). Namun terdapat model ruang warna yang lain seperti YCbCr (luminasi Y dan dua komponen warna Cb dan Cr) yang digunakan pada televisi tabung katoda.

Model warna dapat dibagi menjadi 2 jenis berdasarkan cara pembentukan komponen warnanya yaitu model warna aditif dan model warna substraktif. Model warna aditif adalah model warna yang dibentuk dengan mengkombinasikan energi cahaya dari ketiga warna pokok yang dapat digiring pada warna tertentu jika terjadi dominasi salah satu warna pokok. Sedangkan untuk model warna substraktif, pembentukan warna dilakukan dengan menghilangkan beberapa komponen dari cahaya putih. Komponen warna yang dihilangkan tidak akan berkontribusi dalam pembentukan suatu warna. Contoh dari model ruang warna substraktif adalah CMY(K).

Berikut adalah jenis-jenis ruang warna yang digunakan dalam penelitian Menurut Ebner (2007).

1. RGB

Model ruang warna RGB dijelaskan dengan sebuah kubus. Kubus terdiri dari 3 sumbu, *red, green,* dan *blue* dan ditunjukan pada kubus pada gambar 2.1 yaitu tiap titik dalam kubus menjelaskan sebuah warna unik. Sebuah nilai RGB c = [*r, g, b*] dengan *r, g, b* ∈ [0, 1] menentukan nilai intensitas dari warna *red, green,* dan *blue* yang digunakan untuk menghasilkan sebuah piksel pada layar. Untuk *grayscale* yang ditunjukan dengan garis miring pada ilustrasi kubus di gambar 2.6 berada pada dalam kubus dan dimulai dari warna hitam menuju warna putih.



Gambar 2. 6 Kubus ruang warna RGB (Ebner, 2007)

1. HSI



Gambar 2. 7 Segitiga ruang warna HSI (Ebner, 2007)

Model ruang warna HSI terbagi atas 3 komponen yaitu *Hue, Saturation,* dan *Intensity*. *Hue* mendefinisikan jenis warna yang direpresentasikan dengan sebuah nilai titik. Nilai *hue* adalah nilai sudut diantara vektor dari pusat segitiga menuju titik ujung warna merah dan vektor dari pusat segitiga menuju warna yang diproyeksikan pada segitiga. *Saturation* memberikan informasi jumlah warna putih yang diberikan pada warna murni. *Intensity* memberikan informasi tingkat kecerahan dari titik yang ditunjukan pada tiap piksel. Transformasi HSI dapat diterapkan dengan nilai RGB linear maupun non-linear karena pada dasarnya transformasi HSI hanyalah transformasi sistem koordinat. Gambar 2.7 menunjukan segitiga ruang warna HSI. Transformasi ruang warna HSI dari RGB dilakukan dengan persamaan berikut:

I = (2.7)

H = (2.8)

Dengan  (2.9)

S = (2.10)

1. HSV



Gambar 2. 8 Hexagon ruang warna HSV (Ebner, 2007)

Model ruang warna HSV didefinisikan oleh Smith (1978) dan Watt (2000). Model ruang warna ini diciptakan untuk memodelkan metode seorang artis dalam mencampurkan warna. Maka dari itu model ruang warna HSV paling berguna untuk program ilustrasi atau menggambar. Pada model ruang warna ini terdapat 3 komponen yaitu *hue* (H), *saturation* (S), dan *value* (V). Model ruang warna HSV diilustrasikan sebagaimana kubus ruang warna RGB diorientasikan terhadap sumbu *gray* yang dimulai dari warna hitam menuju warna putih seperti yang diilustrasikan pada gambar 2.8, sehingga membentuk sebuah kerucut hexagon. Nilai *value* menunjukan intensitas maksimum dari komponen warna *red, green,* dan *blue*. Nilai *saturation* menunjukan intensitas minimum dan maksimum stimulus RGB. Intensitas minimum menunjukan seberapa banyak warna putih yang ditambahkan untuk menghasilkan sebuah warna. Persamaan dibawah menjelaskan transformasi dari ruang warna RGB ke ruang warna HSV:

(2.11)

(2.12)

H = (2.13)

1. LAB



Gambar 2. 9 Ruang warna LAB (Agudo dkk., 2013)

Model ruang warna LAB merupakan model ruang warna yang didefinisikan oleh CIE yang terdiri dari *luminance* (L), dan dua *channel* warna (a dan b). Model ruang warna LAB diciptakan sebagai solusi terhadap permasalahan sistem warna XYZ yang menunjukan perbedaan yang besar antara warna hijau dan hijau kekuningan sedangkan perbedaan antara warna merah dan biru sangat kecil. Pada model ruang warna LAB, perbedaan warna yang dipersepsikan sesuai dengan jarak ketika diukur secara kolorimetri. Sumbu a dimulai dari hijau (-a) sampai merah (+a) dan sumbu b dimulai dari biru (-b) sampai kuning (+b). Tingkat kecerahan (L) meningkat dari bawah ke atas dari model 3 dimensi pada gambar 2.9. Model ruang warna LAB cocok untuk tujuan manipulasi citra digital dibandingkan dengan model ruang warna RGB, dan sering digunakan pada program manipulasi gambar seperti penajaman dan menghilangkan objek pada citra JPEG atau dari *scanner* dan kamera digital.

1. YCbCr

Model ruang warna YCbCr dikembangkan sebagai bagian dari ITU-R BT.301 pada pengembangan *world-wide digital component video standard.* Model ruang warna ini sering digunakan pada pengembangan standar JPEG dan digunakan untuk deteksi warna kulit. Persamaan merupakan transformasi ruang warna RGB ke ruang warna YCbCr:

(2.14)

Gambar 2.10 menunjukan kubus warna RGB pada ruang YCbCr dan dapat disimpulkan bahwa tidak semua nilai YCbCr dapat dihasilkan dengan model warna RGB.



Gambar 2. 10 Kubus RGB di dalam ruang warna YCbCr (Ebner, 2007)

## 2.5 Color Constancy



Gambar 2. 11 Objek di bawah daylight (a) dan di bawah skylight (b). (Foster, 2010)

Menurut Geusebroek dkk (2013), *color constancy* adalah suatu fenomena pada persepsi penglihatan manusia yang mengoreksi cahaya warna pantulan dari suatu objek dibawah sumber iluminasi yang berbeda-beda. Dalam konteks fisika dapat dijelaskan, cahaya siang dipantulkan oleh objek dan mencapai mata. Rasio pemantulan dari panjang gelombang energi radiasi merupakan sifat dari obyek, maka dari itu fungsi refleksi dari sebuah buah jeruk merupakan karakteristik fisik dari buah jeruk. Namun, banyaknya energi radiasi yang jatuh pada retina bergantung pada fungsi pemantulan obyek, geometri dari obyek, dan sumber iluminasi. *Color constancy* dapat dijelaskan ketika manusia mempersepsikan sebuah jeruk berwarna oranye baik dibawah cahaya lilin, cahaya matahari, iluminasi yang miring, maupun frontal.

Foster (2010) menjelaskan bahwa, dalam analisis *color constancy* sumber iluminasi tidak diketahui secara langsung oleh pengamat. Hal tersebut merupakan hal yang lumrah, pada situasi ketika sumber cahaya diketahui, pengamatan langsung dengan mata triwarna hanya menghasilkan representasi 3 dimensi dari spektrum. Pada lingkungan alami, sumber iluminasi tidak diketahui secara jelas asal usulnya karena pada umumnya merupakan campuran rumit dari penyinaran secara langsung dan tidak langsung yang terdistribusi dari berbagai sudut insiden, yang kemudian termodifikasi oleh hambatan lokal dan refleksi *mutual*, yang semua komponen tersebut dapat berubah seiring berubahnya waktu dan posisi. Dari keadaan tersebut, diajukan dua sifat dari fenomena *color constancy*. Pertama, dapat memberikan persepsi stabil pada identitas permukaan objek, bersifat independen dari variasi spektrum iluminasi, dan membuat pengamat berinteraksi dengan benar terdap sekitar. Kedua, dapat memungkinkan perkiraan spektrum iluminasi, termasuk pada fase hari terang, dan perkiraan waktu dan cuaca hari. Gambar 2.11 merupakan ilustrasi perbedaan iluminasi pada objek yang sama yang terjadi pada fenomena *color constancy*.

## 2.6 Metode *Inpainting Navier Stokes*

Menurut Bertalmio dkk (2010), metode inpainting adalah proses untuk mengisi data yang hilang pada area yang ditentukan pada citra atau video. Penggunaanya mencakup penghilangan suatu objek dari sebuah citra yang berupa retakan, kotoran, atau elemen sejenis agar citra terlihat diperbaiki dari pengamat. Pada metode *inpainting Navier-Stokes*, proses inpainting melibatkan persamaan *Navier-Stokes* untuk *incompressible fluid* sebagai penentuan arah dari operasi smoothing citra.

Prinsip dasarnya adalah operasi inpainting bergerak mengikuti tepian area yang diketahui menuju ke area yang tidak diketahui (docs.opecv.org, n.d). Algoritma ini meneruskan *isophote* sambil mencocokan vektor-vektor gradien pada batas dari area inpainting. Penentuan *isophote* mengimplementasikan konsep dinamika fluida. Kemudian dilakukan operasi *filling* untuk mengecilkan varian pada area.

## 2.7 Deep Learning

Menurut Kamilaris dan Prenafeta-Boldú (2018) *deep learning* merupakan salah satu metode dari *machine learning* yang menambahkan “kedalaman” (kompleksitas) ke dalam model dan merubah data menggunakan berbagai macam fungsi yang membuat representasi data dengan cara hierarkis melalui beberapa layar abstraksi. Kelebihan dari *deep learning* adalah *feature learning* yaitu ekstraksi fitur otomatis dari data mentah, dengan fitur-fitur dari tingkatan hierarki yang lebih tinggi dibuat dengan komposisi dari fitur-fitur tingkatan yang lebih rendah. Model kompleks yang digunakan dalam *deep learning* memungkinkan untuk paralelisasi masif sehingga tingkat akurasi dapat meningkat dan mengurangi error pada permasalahan regresi. Beberapa jenis model arsitektur *deep learning* adalah *unsupervised pre-trained networks, convolutional neural networks, recurrent neural networks, recursive neural networks*. *Deep learning* dapat digunakan untuk berbagai macam data yaitu citra, video, audio, percakapan, bahasa natural, data kontinyu, dan data diskrit.

## 2.8 CNN

Menurut Nikola (2018), CNN (*Convolutional Neural Network*) merupakan salah satu model *neural network* yang sering digunakan untuk rekognisi citra . Cara kerja CNN meniru cara manusia untuk mengidentifikasi suatu citra objek, yaitu dengan menemukan fitur-fitur unik yang terdapat dalam objek. Pada CNN, fitur-fitur unik dideteksi, pada awalnya fitur-fitur level bawah seperti lekukan dan tepian, kemudian fitur dengan konsep yang lebih abstrak dibangun. Pada dasarnya CNN menggunakan *neural network* standar namun juga menggunakan lapisan-lapisan lain untuk menyiapkan data dan mendeteksi fitur-fitur tertentu. CNN memproses citra melalui beberapa lapisan sebagai berikut.

1. Lapisan Konvolusi



Gambar 2.12 Matriks citra (kiri) dan matriks filter (kanan) (Nikola, 2018)

Lapisan konvolusi berfungsi untuk mendeteksi fitur-fitur pada suatu citra. Fungsi tersebut dilakukan dengan menggunakan banyak filter pada suatu citra yang kemudian mendeteksi atribut tingkat rendah dan tingkat lebih tinggi dari citra. Gambar 2.12 menunjukan kedua matriks yaitu matrik yang diolah sebagai citra dan matrik filter yang digunakan untuk mengolah matriks citra.

Awalnya filter pertama digunakan dan menghasilkan 1 *feature map*, kemudian filter kedua digunakan untuk menghasilkan *feature map* jenis berbeda seperti yang ditunjukan pada Gambar 2.13.



Gambar 2.13 Hasil feature map dari lapisan konvolusi (Nikola, 2018)

1. Lapisan Non-Linearitas

Lapisan non-linearitas merupakan lapisan yang mengubah nilai jumlah dari semua lapisan dalam bentuk fungsi linear menjadi fungsi non-linear. Lapisan ini juga disebut lapisan aktivasi karena menggunakan fungsi aktivasi. Pada umumnya fungsi non-linear yang dipakai adalah *Rectifier Function* (ReLU). Perbedaan citra dari 3 keadaan ditunjukan pada Gambar 2.14.



Gambar 2.14 Gambar citra asli (kiri), gambar citra hasil konvolusi (tengah), dan gambar citra hasil lapisan non-linearitas (kanan) (Nikola, 2018)

Pada gambar *feature map*, nilai warna hitam adalah nilai negatif kemudian setelah digunakan fungsi ReLU maka nilai warna hitam dihilangkan.

1. Lapisan *Pooling* (*Downsampling*)

Lapisan *pooling* berfungsi untuk mengurangi ukuran spasial dari representasi data yaitu jumlah parameter dan komputasi dari jaringan. Pengurangan ukuran matriks pada proses *downsampling* *max pooling* dapat dilihat pada Gambar 2.15.



Gambar 2.15 Proses pooling menggunakan max pooling (Nikola, 2018)

1. Lapisan Pemipihan (*Flattening*)

Lapisan pemipihan adalah lapisan sederhana yang digunakan untuk mempersiapkan data yang menjadi *input* lapisan terhubung penuh. Pada umumnya, *neural network* menerima data dalam bentuk *array* satu dimensi. Maka dari itu, lapisan ini mengubah data dari lapisan *pooling* atau konvolusi menjadi beberapa *array*. Kemudian nilai yang dihasilkan menjadi *input* bagi *neural network*. Perubahan matriks pada proses pemipihan dapat dilihat pada Gambar 2.16.



Gambar 2.16 Array sebelum diproses pada lapisan pemipihan (kiri) dan array setelah mengalami proses pemipihan (kanan) (Nikola, 2018)

1. Lapisan Terhubung Penuh (*Fully Connected Layer*)

Lapisan terhubung penuh merupakan lapisan terakhir yang berfungsi untuk melakukan klasifikasi. Lapisan ini mendapatkan *input* dari lapisan pemipihan yang kemudian data *input* diproses dengan *neural network*.

Menurut Srivastava dkk (2014), dalam implementasi CNN ada permasalahan yang disebabkan oleh jumlah data yang tidak cukup banyak untuk proses latih. Data yang sedikit menyebabkan adanya *sampling noise* yang hanya ada pada *dataset* latih namun tidak ada dalam keadaan sesungguhnya. Fenomena ini disebut *overfitting*. Solusi terhadap fenomena *overfitting* adalah dengan menggunakan *drop out*. *Drop out* mencegah *overfitting* dengan mengeluarkan unit dalam *neural network* secara acak sehingga *neural network* menjadi. *Neural network* yang dirampingkan terdiri dari unit-unit yang lolos dari proses *drop out*.

Menurut Prilianti dkk (n.d.), untuk mengurangi tingkat *error* dalam proses latih menggunakan CNN, fungsi optimasi digunakan. Fungsi optimasi bekerja dengan meminimalkan fungsi *error* saat proses latih kemudian memperbarui parameter internal pada *neural network* .Terdapat berbagai macam fungsi optimasi sebagai berikut.

1. Stochastic Gradient Descent (SGD)

SGD merupakan modifikasi dari metode *gradient descent*. Modifikasi terletak pada jumlah data yang digunakan untuk memperbarui parameter internal (). SGD akan secara perlahan memperbarui parameter internal berdasarkan sebagian dari keseluruhan data. SGD akan menghitung ulang nilai gradien untuk bagian yang lain dan melakukan perbaruan parameter selanjutnya. Perbaruan parameter dilakukan dengan persamaan 2.15. Pada t+1 dengan yaitu *learning rate*, dan *J* sebagai fungsi *error*.

(2.15)

1. Adaptive Gradient (Adagrad)

Pada Adagrad, modifikasi yang dilakukan adalah pada *learning rate* yaitu dapat berubah pada setiap *time step* mempertimbangkan dampak terhadap parameter yang diperbarui. Modifikasi ini dapat memungkinkan tiap parameter untuk menemukan nilai optimal masing-masing secara independen dan meningkatkan akurasi. Persamaan 2.16 menjelaskan perhitungan fungsi optimasi Adagrad dengan yaitu gradien parameter pada fungsi *error* saat waktu *t* dan adalah elemen diagonal dari matrik . Tiap elemen diagonal mewakili jumlah dari kuadrat dari tiap gradien parameter sampai *time step* t.

(2.16)

1. Adaptive Delta (Adadelta)

Adadelta merupakan modifikasi dari Adagrad untuk mengurangi penurunan *learning rate* yang monoton yang disebabkan oleh akumulalsi gradien kuadrat dari tiap *epoch*. Pada Adadelta, yang diakumulasi hanyalah gradien kuadrat w terakhir. Persamaan 2.17 merupakan implementasi akumulasi sebagai *decaying running average* dari gradien kuadrat dengan sebagau konstanta *decay*. Persamaan 2.18 dan persamaan 2.19 merupakan proses pembaruan parameter pada *time step t+1*.

(2.17)

(2.18)

(2.19)

1. Root Mean Squared Propagation (RMSProp)

RMSProp merupakan solusi terhadap permasalahan Adagrad selain Adadelta. RMSProp membagi *learning rate* menggunakan *running average* dari gradien kuadrat yang meluruh secara eksponensial. Persamaan untuk perbaruan parameter ditunjukan dengan persamaaan 2.20 dan 2.21.

(2.20)

(2.21)

1. Adaptive Momentum (Adam)

Adam memiliki kesamaan dengan Adadelta dan RMSProp, yaitu menyimpan gradien kuadrat dari *bias-corrected past epoch* (. Selain itu Adam menumpan nilai *bias-corrected past average gradient* (. Nilai dan digunakan untuk memperkirakan peluruhan momen awal dan momen kedua dari gradien-gradien yang dihitung dengan persamaan 2.22 sampai 2.25 , adalah konstanta peluruhan (*decay*). Kemudian unruk perbaruan parameter pada waktu *t+1* ditunjukan pada persamaan 2.26.

(2.22)

(2.23)

(2.24)

(2.25)

(2.26)

1. Adaptive Max Pooling (Adamax)

Adamax adalah pengembangan dari Adam, modifikasi yang pada penggunaan norma tidak terbatas (. Nilai dalam Adam dengan akan ke nilai yang kebih stabil. Perbaruan parameter pada waktu *t+1* dihitung dengan persamaan 2.7 dan 2.8.

(2.27)

(2.28)

1. *Nesterov Adaptive Momentum* (Nadam)

Nadam merupakan implementasi dari Nesterov Momentum kepada metode Adam. Hipotesa yang ada adalah Nadam akan meraih hasil yang lebih baik dari Adam karena momentum Nesterov dapat meraih hasil yang lebih baik dari momentum klasik. Persamaan 2.29 menjelaskan perbaruan parameter pada waktu *t+1*.

(2.29)

Dalam perancangan arsitektur CNN, performa arsitektur dapat diketahui menggunakan *loss function*. Berikut adalah beberapa jenis *loss function* untuk mengukur performa model CNN untuk tujuan regresi sesuai dengan penelitian ini (Keras.io, n.d.).

1. *Mean squared error*

Persamaan 2.30 Merupakan formula dari *loss function mean squared error* (MSE) dengan sebagai nilai prediksi dan sebagai nilai hasil (Aggarwal, 2018).

(2.30)

1. *Mean absolute error*

Persamaan 2.31 Merupakan formula dari *loss function mean absolute error* (MAE) dengan sebagai nilai prediksi dan sebagai nilai hasil (Guo Hai dkk., 2015).

*MAE =*  (2.31)

1. *Mean absolute percentage error*

Persamaan 2. Merupakan formula dari *loss function mean absolute percentage error* (MAPE) dengan sebagai nilai prediksi dan *y* sebagai nilai hasil (isaacchanghau.github.io, 2017).

(2.32)

1. *Mean squared logarithmic error*

Persamaan 2. Merupakan formula dari *loss function mean squarred logarithmic error* dengan sebagai nilai prediksi dan *y* sebagai nilai hasil (Peltarion.com, n.d.).

(2.33)

## 2.9 Python

Menurut Guttag (2013) Python merupakan bahasa pemrograman *general-purpose*, yaitu bahasa pemrograman yang dapat digunakan untuk membuat hampir segala macam program yang tidak membutuhkan akses langsung ke perangkat keras komputer. Python adalah bahasa pemrograman berbasis objek tingkat tinggi dengan semantik yang dinamis. Struktur data *built-in* tingkat tinggi, dikombinasikan dengan *dynamic typing,* dan *dynamic binding* membuat bahasa pemrograman Python menarik untuk digunakan pada *Rapid Application Development* dan sebagai bahasa *scripting* yang menghubungkan berbagai macam komponen.

Kelebihan Python sebagai bahasa pemrograman adalah kesederhanaan sintaks yang memudahkan *user* untuk mempelajari Python dan adanya *runtime feedback* karena Python didesain untuk diinterpretasikan, dapat membantu programer pemula. Selain itu, tersedia banyak *library* gratis untuk Python yang memberikan berbagai macam fungsionalitas. Sejak diperkenalkannya Python oleh Guido von Rossum pada tahun 1990, Python mengalami banyak perubahan. Pada awalnya Python tidak dikenal dan jarang digunakan, namun semenjak tahun 2000 yaitu munculnya Python 2.0, banyak orang mulai menggunakan bahasa pemrograman Python. Berbagai macam *library* dikembangkan oleh *user* yang dapat terhubung dengan Python menjadikan pengembangan ekosistem Python sebagai kegiatan berbasis komunitas. Pada tahun 2008, Python 3.0 dirilis. Python 3.0 dikembangkan dengan tujuan menghilangkan inkonsistensi pada Python 2.0. Namun banyak *library* yang tidak dapat dijalankan menggunakan Python 3.0.

## 2.10 Flask

Menurut *Flask Documentation* (2017), Flask adalah *micro web framework* yang ditulis dengan Python dan dibuat oleh Armin Ronacher pada tahun 2004. Flask termasuk *microframework* karena tidak memerlukan *tool* dan *library* khusus untuk bekerja. Flask tidak memiliki *database, abstraction layer,*dan *form validation*. Namun, Flask mendukung ekstensi yang dapat menambah fitur aplikasi seakan-akan fitur-fitur diimplementasikan pada Flask. Pada umumnya Flask banyak digunakan dengan MongoDB agar memberikan kendali lebih kepada database.

## 2.11 TensorFlow

Menurut *Tensorflow Guide* (2018), TensorFlow adalah *library* perangkat lunak *open source* untuk komputasi numerik berkinerja tinggi. Arsitektur TensorFlow yang fleksibel memungkinkan penggunaan komputasi lintas platform-platform berbeda (CPU, GPU, TPU) dengan mudah, dan dari desktop menuju kelompok-kelompok server menuju perangkat-perangkat mobile. TensorFlow dikembangkan oleh ilmuwan dan teknisi dari tim Google Brain di dalam organisasi AI Google, yang memiliki dukungan kuat untuk *machine learning* dan *deep learning* dan *core* komputasi numerik fleksibel yang digunakan lintas lingkup ilmu yang lain. TensorFlow adalah *symbolic math library* dan digunakan juga untuk aplikasi *machine learning* seperti *neural network*. TensorFlow adalah sistem *machine learning* yang beroperasi pada skala besar dan *environment* yang heterogen. TensorFlow menggunakan *dataflow graphs* untuk merepresentasikan komputasi, *shared state,* dan operasi-operasi yang merubah *state* dengan memetakan node-node pada *dataflow graph* di berbagai macam mesin dalam sebuah *cluster*, dan di dalam sebuah mesin antar banyak alat komputasi, termasuk *multicore CPU, general purpose GPU,* dan *custom-designed ASIC* yang disebut sebagai *Tensor Processing Unit* (TPU). Arsitektur ini memberikan fleksibiltas kepada *application developer*. TensorFlow mendukung berbagai macam aplikasi dengan fokus pada proses *training* dan *inferensi* pada *deep neural network,* dan telah banyak digunakan untuk riset *machine learning*.

## 2.12 Keras

Menurut Keras.io (n.d.), Keras adalah API *neural network* tingkat tinggi yang ditulis dengan bahasa pemrograman Python. Penamaan “Keras” berasal dari bahasa Yunani yang artinya tanduk. Pada awalnya Keras dikembangkan sebagai bagian dari penelitian proyek ONEIROS (*Open Ended Neuro-Electronic Intelligent Robot Operating System*). Hingga pertengahan tahun 2018, *user* Keras mencapai angka 250.000, menandakan bahwa Keras diadopsi dengan cepat di komunitas ilmuwan dan industri dibandingkan dengan *framework deep learning* lainnya kecuali TensorFlow. Pada proses pengembangannya, Keras didukung oleh Google, Microsoft, Amazon AWS, NVIDIA, Uber, dan Apple. Keras dapat dijalankan menggunakan TensorFlow, CNTK, atau Theano. Keras dikembangkan dengan tujuan untuk memudahkan eksperimentasi yang cepat dan kompatibel dengan Python 2.7 hingga Python 3.6. Keras merupakan *library deep learning* yang:

1. Memungkinkan untuk *prototyping* yang mudah dan cepat melalui modularitas dan ekstensibilitas.
2. Mendukung penggunaan *Convolutional Neural Network* dan *Recurrent Neural Network* termasuk kombinasi keduanya.
3. Berjalan pada CPU dan GPU dengan lancar.

Empat prinsip yang menjadi dasar pengembangan Keras adalah:

1. *User friendliness*

Keras adalah API yang didesain untuk manusia, bukan mesin. Maka dari itu dalam pengembangannya mengutaman *user experience.* Keras mengikuti *best practices* yang menghasilkan API yang sederhana dan konsisten, seperti minimnya aksi *user* yang dibutuhkan untuk penggunaan umum dan memberikan hasil *feedback* yang jelas dan *actionable* ketika ada kesalahan dari *user*.

1. Modularitas

Sebuah model adalah sekuens atau grafik dari *standalone*, modul-modul bisa diatur konfigurasinya sepenuhnya yaitu bisa dipasang satu sama lain dengan batasan yang sesedikit mungkin. Sifat modul tersebut berada pada komponen-komponen Keras seperti *neural layer, cost function, optimizer, initialization schemes, activation function, regularization scheme* yang mana merupakan modul yang berdiri sendiri yang dapat di kombinasikan untuk membuat model-model baru.

1. Ekstensibilitas yang mudah

Modul yang baru dibuat mudah untuk dimodifikasi seperti penambahan *class* dan fungsi. Kemudahan untuk membuat modul-modul baru membuat Keras cocok untuk penelitian maju.

1. Bekerja dengan Python

Semua model dibuat menggunakan bahasa pemrograman Python. Pemakaian Python sebagai bahasa pengembangan Keras disebabkan oleh keuntungan dari Python yaitu *compact,* mudah untuk di*debug*, dan memberikan ekstensibilitas yang mudah.

Aplikasi-aplikasi Keras adalah model-model *deep learning* yang tersedia dengan bobot-bobot yang sudah dilatih sebelumnya. Model-model ini dapat digunakan untuk prediksi, ekstraksi fitur, dan *fine tuning*. Model-model yang tersedia oleh *library* Keras adalah sebagai berikut

1. Xception
2. VGG16
3. VGG19
4. ResNet50
5. InceptionV3
6. InceptionResNetV2
7. MobileNet
8. DenseNet
9. NASNet
10. MobileNetV2

## 2.13 OpenCv

Menurut Bradski dan Kaehler (2012), OpenCv adalah *library computer vision* yang bersifat *open source*. OpenCv ditulis dalam C, dan C++ namun dapat dijalankan menggunakan Python juga dan berjalan pada Linux, Windows, dan Mac OS X. OpenCv didesain untuk efisiensi komputasi dan dengan fokus terhadap aplikasi secara *real time*. OpenCv dapat memanfaatkan keuntungan dari penggunaan prosesor *multicore*. Salah satu dari tujuan OpenCv adalah memberikan infrastruktut *computer vision* yang sederhana untuk digunakan agar pengembang dapat membangun aplikasi *vision* yang rumit dalam waktu cepat. *Library* OpenCv memiliki 500 fungsi dalam banyak bidang *vision*, seperti inspeksi produk pabrik, *medical imaging*, keamanan, *user interface*, kalibrasi kamera, dan robotika.

## 2.14 Numpy

Menurut Oliphant (2016), NumPy adalah *package* fundamental untuk komputasi ilmiah dan pemrosesan *array* tujuan umum yang didesain untuk memanipulasi *array* multi-dimensi besar tanpa mengurangi kecepatan untuk pemrosesan *array* multi-dimensi kecil. Numpy menyediakan beberapa fasilitas yaitu:

1. Objek *array* berdimensi-N yang kuat.
2. Fungsi-fungsi yang canggih.
3. Alat-alat untuk mengintegrasikan kode C/C++ dan Fortran.
4. Fungsi transformasi fourier diskrit, al-jabar linear dasar, dan pembuatan angka *random*

NumPy dibangun berdasarkan dasar kode numerik dan fitur-fitur yang diperkenalkan dengan numarray sebagai perpanjangan dari API C dan mampu untuk menciptakan *array-array* bertipe sementara yang membuat NumPy cocok untuk aplikasi *database* bertujuan umum. Semua NumPy *wheels* yang terdistribusi dari pypi memilki lisensi BSD.

## 2.15 *Flat File*

Menurut Hugo Bowne-Anderson dalam *course* DataCamp *“Importing Data in Python”, flat file* adalah *file* teks data berisi *record* atau data tabel tanpa memiliki hubungan yang tersktruktur, sehingga berbeda sekali dengan *database* relasional yang kolom-kolom pada tabel berbeda dapat berelasi. *Flat file* terdiri dari *record* yaitu suatu baris yang disusun dengan *field,* atau atribut. Ekstensi *file* yang sering dipakai untuk *flat file* adalah .csv atau CSV (*Comma Separated Value)* dan .txt yaitu *file* teks. Nilai-nilai dalam *flat file* dapat dipisahkan dengan karakter-karakter selain koma, yaitu *tab*, dan karakter yang disebut sebagai *delimiter*.

## 2.16 Penelitian Terdahulu

Pada penelitian ini merujuk pada penelitian-penelitian yang telah dilakukan sebelumnya. Namun pada penelitian ini berfokus pada analisa pengaruh berbagai jenis ruang warna terhadap akurasi model CNN yang digunakan untuk mengestimasi kadar pigmen fotosintesis pada citra digital daun, sedangkan pada penelitian yang dirujuk berfokus pada metode pengukuran kadar klorofil dari citra digital daun meggunakan 1 ruang warna saja, yaitu RGB. Berikut ini terdapat beberapa penelitian terhadulu mengenai penggunaan nilai warna pada data citra daun dalam ruang warna RGB untuk mengukur kadar pigmen fotosintesis:

1. Vesali dkk. (2015)

Pada penelitian ini dilakukan pengukuran kadar klorofil dari daun jagung menggunakan citra yang diambil dengan kamera *smartphone*. Berbagai fitur diekstrak dari data RGB citra daun, kemudian digunakan analisa sensitivitas dan *step wise regression* untuk mendapatkan fitur terbaik. Kemudian fitur yang dipilih digunakan sebagai data masukan untuk regresi linear dan model *neural network*. Hasil dari penelitian didapatkan nilai dan RMSE untuk model *linear* sebesar 0.74 dan 6.2, sedangkan untuk model *neural network* sebesar 0.82 dan 5.10. Model yang didapatkan kemudian diimplementasikan pada sebuah aplikasi *smartphone*.

1. Prilianti dkk. (2018)

Prilianti dkk. (2018) telah melakukan penelitian serupa yaitu penggunaan CNN untuk memprediksi pigmen fotosintesis pada tanaman. Penelitian yang dilakukan dengan mengimplementasikan arsitektur CNN yang diberi data input berupa nilai RGB dari citra tanaman dengan jumlah data citra yaitu 391. Arsitektur CNN dilatih untuk memprediksi kadar klorofil dari tanaman dalam citra yang data targetnya adalah nilai kadar klorofil. Nilai kadar klorofil, karotenoid, dan antosianin didapatkan dari persamaan 2.34 sampai dengan 2.38 dengan nilai R adalah nilai intensitas cahaya pada panjang gelombang tertentu yang dipantulkan oleh daun tanaman. Persamaan 2.34 dan 2.35 adalah rumus perhitungan kandungan klorofil. Persamaan 2.36 dan 2.37 adalah rumus perhitungan kandungan karotenoid. Persamaan 2.38 adalah rumus perhitungan kandungan antosianin.

(2.34)

(2.35)

(2.36)

(2.37)

(2.38)



Gambar 2. 17 Arsitektur ShallowNet (Prilianti dkk., 2018)



Gambar 2. 18 Arsitektur LeNet (Prilianti dkk., 2018)



Gambar 2. 19 Arsitektur AlexNet (Prilianti dkk., 2018)

Arsitektur CNN yang digunakan dalam penelitian ini ada tiga yaitu, *Shallownet, LeNet,* dan *AlexNet*. Gambar 2.17 menunjukan ilustrasi desain arsitektur *ShallowNet,* Gambar 2.18 menunjukan ilustrasi desain arsitektur *LeNet,* danGambar 2.19 menunjukan ilustrasi desain arsitektur *AlexNet*. Indikator kinerja arsitektur CNN yang digunakan adalah *Mean Squared Error* (MSE) sebagaimana dijelaskan dengan persamaan 2.32 , dengan adalah nilai yang dihasilkan CNN dan n adalah jumlah data.

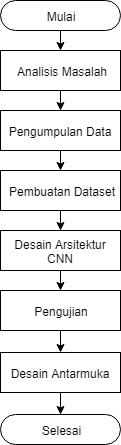
(2.39)

Penelitian yang dilakukan menyimpulkan bahwa arsitektur terbaik yang digunakan untul memprediksi kandungan pigmen fotosintesis adalah *ShallowNet* dengan *optimizer* Adam, ukuran batch 15, dengan 15 *epoch* latih. Arsitektur terbaik menghasilkan nilai MSE pelatihan sebesar 0.0055 dan MSE pengujian sebesar 0.029 dengan rentang 0.1 sampai dengan 2.2.

# Bab III

**Analisis dan Perancangan Sistem**

Alur dalam penelitian untuk menganalisa dampak jenis ruang warna pada akurasi estimasi kadar pigmen menggunakan CNN terdiri dari beberapa tahap yaitu, analisis masalah, pengumpulan data citra daun beserta kandungan klorofilnya, pembuatan dataset dengan mengolah data citra mentah yang didapatkan, desain arsitektur CNN yang digunakan untuk proses latih dan prediksi dengan menggunakan data citra daun, pembuatan program, pengujian akurasi CNN terhadap ruang warna berbeda, dan pembuatan antarmuka. Tahapan penelitian diilustrasikan pada Gambar 3.1.



Gambar 3. 1 Alur penelitian

## 3.1 Analisis Masalah

Metode pengukuran pigmen fotosintesis pada tanaman terbagi menjadi dua jenis yaitu destruktif dan non-destruktif. Kedua metode dapat dicapai menggunakan alat spektrofotometer yaitu untuk metode destruktif menggunakan spektrofotometer UV-Vis yang objek analisanya harus dihancurkan terlebih dahulu, dan spektrofotometer optik untuk metode non-desktruktif. Untuk metode non-destruktif, cara selain menggunakan spektrofotometer adalah dengan menggunakan analisa citra digital menggunakan program komputer. Pengukuran kadar pigmen fotosintesis menggunakan citra digital dapat menjadi metode yang lebih murah dan cepat jika dibandingkan dengan metode lain sehubungan dengan banyaknya *user* *smartphone* dan *personal computer*.

Pengukuran kadar pigmen menggunakan citra digital melibatkan data warna dari citra sehingga pemilihan ruang warna memiliki dampak pada proses pengolahan data yang disebabkan oleh perbedaan nilai dari tiap-tiap ruang warna. Selain itu, perbedaan pola nilai warna pada citra dapat terjadi karena fenomena *color constancy* dan fenomena refleksi cahaya putih dari permukaan daun. Hipotesis yang akan diuji adalah jenis ruang warna selain RGB dapat memberikan pola yang konsisten terhadap fenomena *color constancy*. Maka dari itu dilakukan analisa dampak dari ruang warna berbeda terhadap akurasi model CNN untuk mengestimasi kandungan pigmen di dalam daun. Selain itu dilakukan *preprocessing* untuk data citra daun yang memiliki refleksi cahay putih pada permukaan.

## 3.2 Pengumpulan Data

Data yang dikumpulkan tidak diperoleh pada penelitian ini namun menggunakan data yang digunakan pada penelitian oleh Prilianti (2018). Data yang dikumpulkan adalah 212 data citra daun beserta kandungan pigmen fotosintesis dari tiap-tiap objek daun pada citra. Tanaman yang digunakan dalam pengumpulan data citra daun adalah *piper betle, jasminum, syzigium oleina,* dan *graphtophyllum pictum*. Metode yang digunakan untuk perolehan data dibagi menjadi 2 langkah. Langkah pertama adalah akuisisi data citra daun menggunakan kamera digital *smartphone*. Langkah kedua adalah pengukuran kadar pigmen fotosintesis. Pengukuran kadar pigmen fotosintesis melalui tahap persiapan sebelum dimasukan pada spektrofotometer UV-vis. Tahap persiapan yaitu pembagian bagian daun menjadi 2 bagian yang kemudian dihancurkan menjadi bagian kecil dengan berat masing-masing sebesar 0.05 gram. Bagian pertama untuk pengukuran kadar klorofil dan karotenoid, sedangkan bagian kedua untuk pengukuran kadar antosianin. Larutan yang digunakan untuk ekstraksi pigmen dibedakan dari jenis pigmennya. Untuk klorofil dan antosianin, pelarut mengandung 100% aseton, sedangkan untuk antosianin pelarut merupakan campuran dari metanol, asam hidroklorik terkonsentrasi, dan air saringan. Pelarut sebanyak 1.5 mL kemudian ditambahkan kalsium karbonat dan sodium askorbat dan dimasukan ke dalam tabung. Larutan dan sampel daun kemudian dicampur menggunakan *vortex* selama 1 menit, dan didinginkan selama 1 menit selama 3 kali untuk tiap tabung, kemudian disentrifugasi dengan kecepatan 14,000 rpm selama 2 menit dan kemudian didinginkan. Setelah tahap persiapan selesai, sampel diukur nilai absorbansinya menggunakan spektrofotometer dan dihitung kadar pigmennya menggunakan metode Lichtentaler (1983) dan metode Sims dan Gammon (2002). Gambar 3.2 menunjukan data citra daun yang diambil menggunakan kamera digital *smartphone*. Tabel 3.1 menunjukan contoh data kadar pigmen fotosintesis sesuai dengan kode citra daun.



Gambar 3.2 Data citra sampel daun

Tabel 3.1 Data kadar pigmen fotosintesis

|  |  |  |  |
| --- | --- | --- | --- |
| Kode citra | Antosianin (Abs/g) | Karotenoid (g/g) | Klorofil (g/g) |
| G01 | 25.8532 | 0.0000 | 461.9024 |
| J01 | 0.0000 | 0.0000 | 1011.7251 |
| PB01 | 0.7897 | 11.6864 | 102.6544 |
| PM01 | 0.0296 | 8.8892 | 357.2901 |

## 3.3 Pembuatan Dataset

Tabel 3.2 Data kadar pigmen fotosintesis ternormalisasi

|  |  |  |  |
| --- | --- | --- | --- |
| Kode citra | Antosianin (Abs/g) | Karotenoid (g/g) | Klorofil (g/g) |
| G01 | 0.2881 | 0.0000 | 0.4565 |
| J01 | 0.0000 | 0.0000 | 1.0000 |
| PB01 | 0.0088 | 0.2195 | 0.1015 |
| PM01 | 0.0003 | 0.1670 | 0.3531 |

Pembuatan dataset dilakukan pada dua jenis data yaitu data pengukuran kadar pigmen fotosintesis dan data mentah citra daun. Data pengukuran kadar pigmen fotosintesis yang didapat dirubah nilainya dengan melakukan operasi normalisasi untuk memperkecil *magnitude* datanya. Data kadar pigmen yang dinormalisasi ditunjukan dengan Tabel 3.2. Data mentah citra yang diperoleh dari tahap pengumpulan data masih memerlukan tahap *preprocessing* agar menjadi dataset yang sesuai. Proses *preprocessing* data terbagi menjadi 3 yaitu, penghilangan refleksi cahaya putih pada permukaan, pembuatan *channel-channel* ruang warna RGB, HSV, LAB, dan YCbCr, dan kemudian proses augmentasi data yang kemudian menjadi dataset *input* yang dimasukan pada CNN. Penggunaan dataset dalam proses latih model CNN yaitu 80% data untuk data latih dan 20% untuk data validasi. Penjelasan tahap pembuatan dataset adalah sebagai berikut:

1. Penghilangan efek refleksi cahaya putih pada permukaan

Refleksi cahaya putih diminimalisir menggunakan algoritma Alexandru Telea dan Navier Stokes. Kedua algoritma tersebut tesedia dalam bentuk fungsi inpaint yang ada pada *library* OpenCv. Operasi inpaint dilakukan disertai dengan elemen *mask* yang agar operasi inpaint tepat dilakukan pada bagian refleksi cahaya putih pada citra daun. Langkah-langkah operasi peniadaan refleksi cahaya putih pada permukaan daun adalah sebagai berikut:

* + 1. Akses data citra daun.
    2. Segmentasi objek daun.
    3. Pembuatan *mask* untuk target posisi refleksi cahaya putih pada citra daun.
    4. Operasi inpaint.

1. Pembuatan *channel-channel* tiap ruang warna

Pembuatan *channel-channel­* ruang warna ditempuh dengan konversi ruang warna RGB menjadi ruang warna HSV, LAB, dan YCbCr. Setelah operasi konversi ruang warna dilakukan, komponen ruang warna akan dipisahkan dan disimpan masing-masing sebagai komponen independen. Dari proses yang dilakukan terbentuk dataset komponen-komponen dari berbagai ruang warna yang dapat dikombinasikan sebagai dataset yang dimasukan pada model CNN.

* + 1. Akses data citra daun.
    2. Konversi ruang warna.
    3. Pemisahan channel.
    4. Penyimpanan file channel.

1. Augmentasi data

Proses augmentasi data dilakukan untuk mengatasi kemungkinan performa CNN yang rendah karena sedikitnya dataset yang tersedia. Augmentasi data yang dilakukan merupakan penambahan variasi data dengan memutar data dengan berbagai sudut putaran, memperbesar, dan diterapkan standarisasi. Data yang dihasilkan dari 2 proses sebelumnya, diterapkan operasi augmentasi data.

## 3.4 Desain Arsitektur CNN

Penelitian ini berfokus pada analisa efek yang dihasilkan dari penggunaan jenis ruang warna berbeda terhadap akurasi CNN, maka dari itu arsitektur yang digunakan adalah arsitektur CNN dari penelitian terdahulu dari (Prilianti, 2018). Berikut adalah deskripsi rancangan arsitektur yang digunakan dengan Tabel 3.3 dan diilustrasikan dengan Gambar 3.3.

Tabel 3. 3 Arsitektur Model CNN

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Layer | | Feature Map | Size | Kernel Size | Stride | Activation | Dropout |
| Input | Citra |  |  |  |  |  |  |
| 1 | *Convolution* | 32 | input | 3x3 | 2x2 | ReLU | - |
| 2 | *Convolution* | 32 |  | 3x3 | 1x1 | ReLU | - |
| 3 | *Pooling* | - |  | 3x3 | 2x2 |  | - |
| 4 | *Flattening* | - |  | - | - |  | - |
| 5 | *Dense* | - | 90 | - | - | Sigmoid | 0.03 |
| 6 | *Dense* | - | 3 | - | - | LeakyRelu  (alpha=0.3) | - |

Arsitektur CNN pada penelitian ini menggunakan *mean squared error* sebagai *loss function* dan *Nesterov Adam* sebagai fungsi optimasi. Secara teknis model CNN dibuat menggunakan *library* Keras dan bahasa Python. *Input* CNN adalah data citra daun dari berbagai macam ruang warna dan *output* yang dihasilkan adalah estimasi kandungan pigmen fotosintesis yaitu klorofil, karotenoid, dan antosianin.



Gambar 3.3 Arsitektur model CNN

## 3.5 Perancangan Desain Sistem

Pada subbab ini dijelaskan rancangan sistem yang membahas tentang urutan alur kerja sistem dan rancangan *file* penyimpanan data. Penjelasan rancangan sistem diberikan dalam bentuk *flowchart, sequence diagram, use case diagram,* dan desain *flat file* database yang digunakan. Penjelasan alur kerja sistem mencakup beberapa fungsi pada aplikasi web yaitu, fungsi *sign up,* fungsi *delete user,* fungsi *sign in*, fungsi latih model, fungsi estimasi pigmen, serta fungsi lihat dan hapus *record* data latih.

## 3.5.1 Flowchart Alur Kerja Sistem

1. *Sign up*



Gambar 3.4 Flowchart alur kerja fungsi sign up

Fungsi *sign up* dimulai dengan *user* memasukan *username* dan *password* yang diinginkan sebagai data input dan memencet tombol *sign up.* Kemudian server akan memeriksa jika data sudah tersedia atau belum pada *flat file user*. Jika belum tersedia maka data *username* dan *password* akan tersimpan pada *flat file user*, jika tidak maka akan muncul pesan yang menunjukan proses *sign up* gagal dan *user* dapat mengulangi tahapan yang sama di awal jika ingin melakukan *sign up*. *User* yang telah terdaftar dapat mengakses fungsi latih model, fungsi estimasi pigmen, serta fungsi lihat dan hapus *record* data latih dengan *username* dan *password* yang telah berhasil terdaftar pada proses *sign up*. Gambar 3.4 menunjukan *flowchart* alur kerja fungsi *sign up.*

1. *Delete* *user*

Fungsi *delete user* adalah fungsi yang memungkinkan *admin* untuk menghapus *user* yang terdaftar pada sistem. Fungsi *delete user* dimulai dengan sistem menampilkan daftar *user* yang terdaftar dengan mengambil data dari *flat file user*. Kemudian *admin* memencet tombol hapus yang berada disamping nama *user* pada table daftar *user*. Kemudian sistem akan meng-*overwrite* *flat file user* dengan data *user* tanpa *user* yang dihapus oleh *admin*. Jika proses hapus data sukses maka akan muncul pesan bahwa penghapusan *user* berhasil, jika tidak maka akan ditampilkan pesan proses hapus *user* gagal. Gambar 3.5 menunjukan *flowchart* alur kerja fungsi *delete user*.



Gambar 3.5 Flowchart alur kerja fungsi delete user

1. Sign in

Fungsi *sign in* berfungsi agar *user* dapat masuk sebagai *user* yang terdaftar pada sistem dan memiliki level otentikasi yang dapat mengakses fungsi-fungsi lain pada sistem. Fungsi ini dimulai dengan *user* memasukan data *username* dan *password*, kemudian *user* memencet tombol “*Sign in”*. Tahapan selanjutnya sistem akan mencocokan data *username* dan *password* dengan data pada *flat file user*, jika ditemukan dan cocok maka *user* akan diarahkan pada halaman beranda. Jika tidak maka akan muncul pesan yang menunjukan bahwa proses *sign in* gagal. Jika *user* ingin mencoba kembali maka mengulang dari tahapan awal. Gambar 3.6 menunjukan *flowchart* alur kerja fungsi *sign in*.



Gambar 3.6 Flowchart alur kerja fungsi sign in

1. Latih model

Gambar 3.7 *Flowchart* alur kerja fungsi latih model

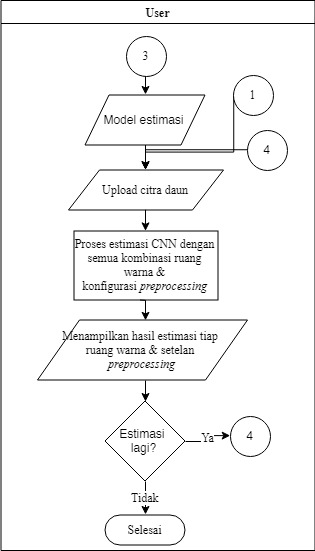
Fungsi latih model berfungsi agar *user* dapat melakukan proses latih model CNN dan menyimpan hasil latihnya. Fungsi ini dapat diawali dengan keputusan *user* untuk membuat *project* baru atau memilih *project* lama. Jika membuat *project* baru, *user* memasukan nama *project* kemudian memencet tombol “*Create*” dan sistem akan menyimpan *project* pada *flat file project* jika nama *project* belum tersedia, jika tidak maka akan muncul pesan bahwa nama *project* sudah tersedia. Jika *user* memilih untuk memakai *project* lama, maka akan tersedia daftar *project* yang sudah dibuat yang datanya diambil dari *flat file* project dan ditampilkan dalam bentuk tabel oleh sistem. Setekah masuk pada *project*,sistem akan menampilkan opsi ruang warna yang dapat dipilih *user*. Setelah ruang warna dipilih dan tombol “Latih” dipencet oleh *user*, proses latih dimulai. Setelah proses latih selesai, sistem menampilkan informasi nilai akurasi hasil latih. *User* dapat memilih untuk menyimpan data nilai akurasi latih dan dapat memilih untuk tidak menyimpan. Jika *user* memilih menyimpan, maka *user* dapat memencet tombol “Simpan”. Jika proses penyimpanan nilai akurasi pada *flat file* ruang warna sukses, maka akan muncul pesan bahwa penyimpanan sukses. Jika *user* memilih untuk melakukan proses latih lagi, *user* dapat kembali ke tahap pemilihan ruang warna. Gambar 3.7 menunjukan *flowchart* alur kerja fungsi latih model.

1. Estimasi pigmen

Fungsi estimasi pigmen berfungsi untuk melakukan proses estimasi kadar pigmen fotosintesis pada citra daun. Fungsi ini dapat dimulai sesuai dengan keputusan *user* untuk merubah model CNN yang digunakan untuk estimasi. Jika *user* memilih untuk merubah model yang digunakan, *user* dapat memencet tombol “Edit Model”.



Gambar 3.8 Flowchart alur kerja fungsi estimasi pigmen



Gambar 3.9 Flowchart alur kerja fungsi estimasi pigmen (lanjutan)

Setelah tombol ”Edit Model” dipencet, sistem akan mengambil data model CNN yang saat ini dipakai untuk estimasi dari *flat file model\_selected* kemudian data akan ditampilkan dalam bentuk tabel. Kemudian tombol “Rubah” pada tabel sebaris dengan ruang warna yang akan dirubah modelnya dipencet. Kemudian sistem akan menampilkan daftar *project* yang memiliki model untuk ruang warna yang telah dipilih. Setelah *user* memencet tombol “Pilih” disamping keterangan *project* yang menyediakan model CNN, sistem akan menampilkan daftar model CNN yang tersedia dari *project* yang telah dipilih. Kemudian *user* dapat memencet tombol “Pilih” yang sejajar dengan model yang akan dipilih untuk proses estimasi kadar pigmen fotosintesis dari citra. Setelah *user* selesai merubah model CNN, *user* dapat memencet tombol “Selesai”. Selanjutnya *user* mengunggah citra daun dan memencet tombol “Estimasi”. Pada tahap ini sistem akan mengestimasi kadar pigmen menggunakan model CNN yang digunakan tiap ruang warna. Setelah proses estimasi selesai, sistem akan menampilkan nilai estimasi kadar pigmen fotosintesis pada daun tiap ruang warna. Gambar 3.8 dan 3.9 menunjukan *flowchart* alur kerja fungsi estimasi pigmen.

1. Lihat dan hapus *record* data latih

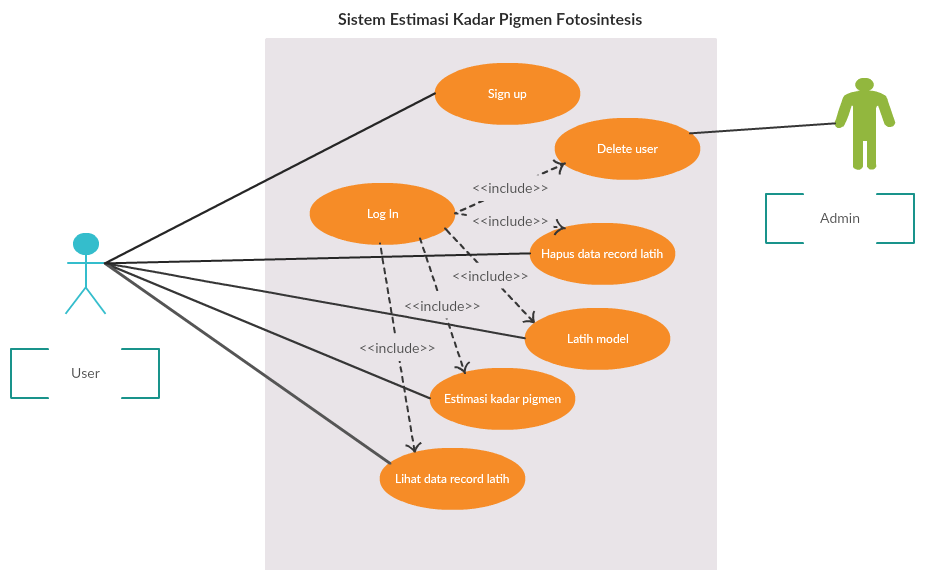


Gambar 3.10 Flowchart alur kerja fungsi lihat dan hapus data record

Fungsi ini merupakan fungsi yang membuat *user* dapat melihat data nilai akurasi latih model yang pernah disimpan pada *project*. Fungsi diawali dengan *server* mengambil data *project* dari *flat file project* kemudian ditampilkan kepada *user*. Setelah *user* memilih *project, server* mengambil data nilai akurasi dari *flat file* ruangwarna berdasarkan *project* yang dipilih lalu *server* membuat data grafik berdasarkan data nilai akurasi yang diambil. Kemudian data nilai akurasi dan grafik nilai akurasi ditampilkan kepada *user*. *User* memiliki pilihan untuk menghapus nilai akurasi dengan memencet tombol “Hapus” yang sejajar pada baris nilai akurasi. Jika *user* memilih menghapus maka *server* akan melakukan penghapusan dengan proses *overwriting* pada *flat file* ruangwarna. Kemudian *server* akan mengambil lagi data nilai akurasi dari *flat file* ruang warna dan mengolahnya untuk ditampilkan kembali pada *user.* Selain itu, *user* dapat mengunduh rekap data nilai akurasi. Jika *user* memilih mengunduh dengan memencet tombol “Download” maka *server* akan memberikan *file* rekap data nilai akurasi dalam bentuk CSV. Gambar 3.12 menunjukan *flowchart* alur kerja fungsi lihat dan hapus data *record.*

## 3.5.2 Use Case Diagram Sistem

Terdapat dua aktor pada aplikasi web estimasi kadar pigmen fotosintesis yaitu, *user* dan *admin*. *User* dapat mengakses fungsi *sign in*, fungsi *sign up,* fungsi latih model, fungsi latih model, fungsi estimasi kadar pigmen, fungsi hapus data *record* latih, fungsi lihat data *record* latih. *Admin* dapat mengakses fungsi *delete user*, dan *sign in*. Setiap aktor harus melalui proses *sign in* dahulu untuk mengakses fungsi-fungsi pada aplikasi. Gambar 3.11 menunjukan *use case* diagram sistem estimasi kadar pigmen fotosintesis.



Gambar 3.11 Use case diagram sistem estimasi kadar pigmen fotosintesis

## 3.5.3 Sequence Diagram Sistem

1. Sign in

Pada tahap awal proses sign in, *user*/*admin* melakukan permintaan login dengan memasukan *username* dan *password,* serta menekan tombol “Masuk”. Kemudian server mengecek *username*, dan *password* yang dimasukan dengan data user. Setelah itu, server akan menyetel status proses *sign in, username*, dan level otentifikasi. Jika status proses *sign in* bernilai “sukses” maka akan muncul tampilan pesan sukses. Jika status proses *sign in* bernilai “gagal” maka akan muncul tampilan pesan sukses. Gambar 3.12 menunjukan *sequence diagram* fungsi *sign in*.



Gambar 3.12 Sequence diagram fungsi sign in

1. *Delete* *user*



Gambar 3.13 Sequence diagram fungsi delete user

Pada tahap awal *delete user*, *admin* menekan tombol “Hapus *user*” yang sejajar pada *username* yang akan dihapus. Kemudian *server* akan mencari data *user* yang akan dihapus pada *flat file user* berdasarkan *id user*. Jika ditemukan maka akan menghapus baris data yang terdiri dari data *id user, username, password,* dan *level otentifikasi* sesuai dengan *user* yang dipilih oleh *admin*. Kemudian server akan menyetel status *delete* dan daftar data *user*. Jika status *delete* sukses maka akan menampilkan pesan penghapusan *user* sukses dan memperbarui tampilan daftar *user* yang tersedia. Gambar 3.13 menunjukan *sequence diagram* fungsi *delete user*.

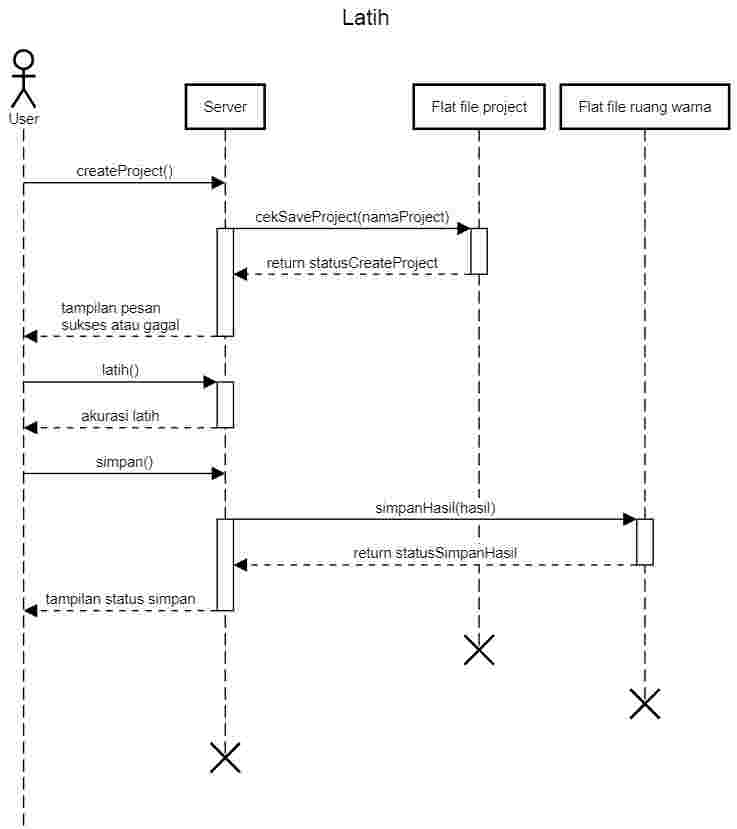
1. *Sign up*

Pada tahap awal *sign up*, *user* memasukan data *username* dan *password* serta memencet tombol *sign up.* Kemudian *server* akan mengecek data pada *flat file* *user* untuk memastikan *username* sudah tersedia atau belum. Jika sudah tersedia, status sukses akan disetel “gagal”. Jika belum tersedia, data *user* akan disimpan pada *flat file user* dan status sukses akan disetel “sukses”. Setelah itu, akan muncul tampilan status proses *sign up.* Gambar 3.14 menunjukan *sequence diagram* fungsi *sign up*.



Gambar 3.14 Sequence diagram fungsi sign up

1. Latih model



Gambar 3.15 Sequence diagram fungsi latih model

Pada tahap awal latih model, *user* membuat *project* baru dengan menekan tombol *new project* dan memasukan nama *project*. Kemudian *server* akan mengecek jika nama *project* sudah tersedia atau belum pada *flat file project.* Jika sudah tersedia, akan ditampilkan status yang menunjukna bahwa nama *project* yang dimasukan sudah tersedia, jika belum maka nama *project* yang dimasukkan akan disimpan pada *flat file project* lalu diarahkan pada tampilan opsi pelatihan model CNN. Setelah *user* memilih pilihan ruang warna dan *preprocessing* serta memencet tombol latih, *server* melakukan proses latih. Setelah proses latih selesai maka akan muncul tampilan yang menunjukan nilai akurasi hasil latih. *User* dapat menyimpan hasil nilai akurasi ketika *user* memencet tombol simpan. Setelah tombol simpan dipencet maka server akan menyimpan nilai akurasi pada *flat file* ruang warna sesuai dengan jenis ruang warna yang dipilih pada opsi latih. Setelah itu akan ditampilkan pesan bahwa penyimpanan telah sukses dilakukan. Gambar 3.15 menunjukan *sequence diagram* fungsi latih model.

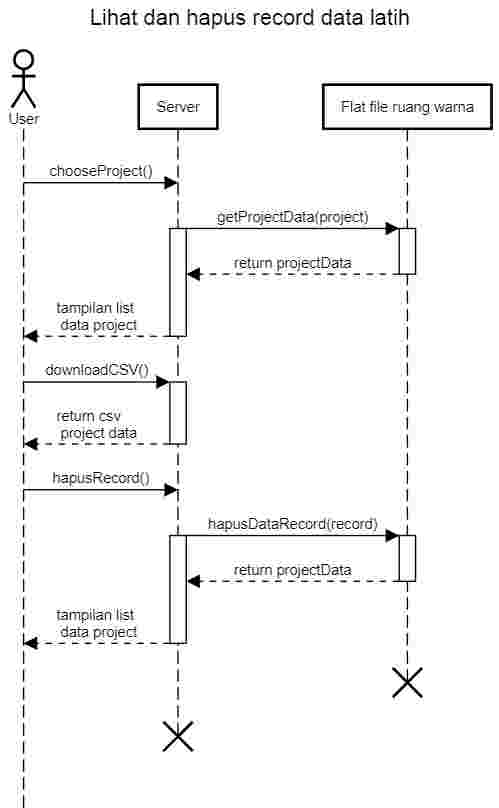
1. Estimasi pigmen



Gambar 3.16 Sequence diagram fungsi estimasi kadar pigmen

Pada tahap awal estimasi pigmen, *user* dapat merubah file model masing-masing ruang warna yang digunakan untuk mengestimasi pigmen dari citra daun. Untuk merubah file model, *user* harus memilih file model yang tersedia pada warna tertentu. Setelah model dipilih oleh *user*, *server* akan melakukan *overwrite* pada *flat file model\_selected* sehingga berisi data nama file model yang dipilih oleh *user* untuk digunakan pada proses estimasi. Setelah *user* tidak perlu untuk melakukan perubahan pada file model yang digunakan, *user* dapat memencet tombol “Selesai”. Kemudian, untuk melakukan proses estimasi, *user* dapat mengunggah citra daun dengan memencet tombol “Pilih file” lalu memencet tombol “Estimasi”. Setelah data citra dimasukkan, *server* melakukan proses estimasi menggunakan 20 model dari kombinasi beberapa ruang warna dan setelan *inpaint*. Ketika proses estimasi selesai, maka *server* akan menampilkan nilai estimasi kadar pigmen dari citra daun berdasarkan masing-masing jenis ruang warna dan *preprocessing* yang digunakan. Gambar 3.16 menunjukan *sequence diagram* fungsiestimasi pigmen.

1. Lihat dan hapus *record* data latih



Gambar 3.17 Sequence diagram fungsi lihat dan hapus record data latih

5Untuk melihat *record* data latih dari sebuah *project,* pada tahap awal *user* memilih *project* yang tersedia. Kemudian server akan mengambil data nlai akurasi latih berdasarkan nama *project* yang dipilih dari semua *flat file* ruang warna. Setelah pengambilan data nilai akurasi selesai, data akan ditampilkan dalam bentuk yang informatif. *User* dapat mengunduh file CSV rekap data nilai akurasi dari *project* dengan memencet tombol “Unduh”. Setelah tombol “Unduh” dipencet, maka server akan mengembalikan file CSV rekap data nilai akurasi *project* yang dipilih kepada *user*. Selain unduh, *user* dapat menghapus data nilai akurasi pada *project* yang dipilih. Saat *user* memencet tombol “Hapus” yang sejajar dengan nilai akurasi yang akan dihapus, server akan menghapus data nilai akurasi yang dipilih dengan melakukan *overwrite* pada *flat file* ruang warna menggunakan data yang tidak menyertakan nilai akurasi yang dipilih *user*  untuk dihapus. Gambar 3.17 menunjukan *sequence diagram* fungsi lihat dan hapus *record* data latih.

## 3.5.4 Desain Flat File Database

Pada sub-bab ini dijelaskan keterangan dari nilai data pada *flat file* yang digunakan untuk penyimpanan data. Penjelasan mencakup keterangan *field* dari baris data pada *flat file* serta contoh datanya. Indeks angka pada tabel bertujuan untuk menunjukan urutan *field* pada baris data *flat file.* Perlu diketahui bahwa, keterangan *field* bukan berarti nama kolom yang benar-benar tercantum pada *flat file*.

1. model\_select

*Flat file model\_select* merupakan *flat file* yang berisi data yang dibutuhkan untuk mengambil file model yang digunakan pada proses estimasi dari direktori yang tepat. Jumlah baris *flat file model­\_select* bersifat tetap yaitu sejumlah 20 baris, masing-masing baris mewakili penggunaan kombinasi ruang warna tertentu dan *preprocessing* pada proses estimasi. *Field* pertama merupakan ruang warna yang berisi jenis ruang warna, *field* kedua adalah *inpaint* yang berisi status *inpaint, field ketiga* adalah nama *project* yang berisi nama *project* dari model yang telah dipilih untuk proses estimasi, *field* keempat adalah nama model, *field* kelima adalah *timestamp* yang berisi nilai detail waktu model yang dipilih untuk proses estimasi dibuat. Tabel 3.4 menunjukan keterangan tiap *field* dari *flat file model\_select*, sedangkan Tabel 3,5 menunjukan contoh dari data *flat file model\_select*.

Tabel 3. 4 Keterangan field flat file model\_select

|  |
| --- |
| **model\_select** |
| 1. Ruang warna 2. *Inpaint* 3. Nama *project* 4. Nama model 5. Timestamp |

Tabel 3. 5 Contoh data flat file model\_select

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| rgb | yes | experiment\_06 | model\_inpaint\_rgb | 2019:06:11\_00:24:21.417541 |
| hsv | no | trial-01 | model\_hsv | 2019:06:04\_15:29:06.256473 |

1. project

*Flat file project* berfungsi untuk menyimpan *project* yang telah dibuat oleh *user. Field* pertama adalah nama *project* yaitu berisi nama *project* yang telah dibuat, *field* kedua adalah *project creator* yang berisi *username* dari *user* yang membuat *project.* Tabel 3.6 menunjukan keterangan tiap *field* dari *flat file project*, sedangkan Tabel 3.7 menunjukan contoh dari data *flat file project.*.

Tabel 3.6 Keterangan field flat file project

|  |
| --- |
| **project** |
| 1. Nama *project* 2. *Project* *creator* |

Tabel 3.7 Contoh data flat file project

|  |  |
| --- | --- |
| experiment-11 | adn |

1. user

*Flat file user* berfungsi untuk menyimpan data akun *user* dan *admin* yang memungkin *user* atau *admin* untuk melakukan proses *sign in*. *Field* pertama merupaka *id user* yang menyimpan nilai *id* dari akun *user. Field* kedua merupakan *username* yang berisi nilai nama *user. Field* ketiga yaitu *password* berisi nilai *password* yang sudah terenkripsi menggunakan algoritma MD5. *Field* keempat adalah level otentifikasi yang berisi nilai level otentifikasi untuk menentukan jika akun adalah *user* atau *admin*. Tabel 3.8 menunjukan keterangan tiap *field* dari *flat file user*, sedangkan Tabel 3.9 menunjukan contoh dari data *flat file user*.

Tabel 3.8 Keterangan field flat file user

|  |
| --- |
| **user** |
| 1. *id* *user* 2. *username* 3. *password* 4. level otentifikasi |

Tabel 3.9 Contoh data flat file user

|  |  |  |
| --- | --- | --- |
| 1 | adn | f5990f806ce19f39b9c2329e137a4565 |
| 2 | master | 803c14aec7a828e3e1c08ef981da76c9 |

1. ruang warna

*Flat file* ruang warna berfungsi untuk menyimpan nilai akurasi hasil latih yang kemudian menjadi informasi mengenai tingkat akurasi model dengan ruang warna dan *preprocessing* tertentu. Sesungguhnya *flat file* ruang warna terdiri dari 20 *file* sesuai dengan kombinasi ruang warna dan *preprocessing*  (rgb, rgb\_hsv, rgb\_lab, rgb\_ycbcr, hsv, hsv\_lab, hsv\_ycbcr, lab, lab\_ycbcr, ycbcr, rgb\_inpaint, rgb\_hsv\_inpaint, rgb\_lab\_inpaint, rgb\_ycbcr\_inpaint, hsv\_inpaint, hsv\_lab\_inpaint, hsv\_ycbcr\_inpaint, lab\_inpaint, lab\_ycbcr\_inpaint, ycbcr\_inpaint), namun untuk keterangan *field* pada baris data *flat file* memiliki struktur dan jenis nilai yang sama. *Field* pertama adalah nama *project* yang berisi nilai nama *project* yang dipilih *user* ketika melakukan proses latih. *Field* kedua adalah *train loss* yang berisi nilai MSE latih. *Field* ketiga adalah *train* MAE yang berisi nilai MAE latih. *Field* keempat adalah *validation loss* yang berisi nilai MSE validasi. *Field* kelima adalah *validation* MAE yang berisi nilai MAE validasi. *Field* keenam adalah *timestamp*  yang berisi nilai detail waktu proses latih dilakukan. *Field* ketujuh adalah nama *trainer* yang berisi nilai *username user* yang melakukan proses latih. Tabel 3.10 menunjukan keterangan tiap *field* dari *flat file* ruang warna, sedangkan Tabel 3.11 menunjukan contoh dari data *flat file* ruang warna.

Tabel 3.10 Keterangan field flat file ruang warna

|  |
| --- |
| **warna** |
| 1. Nama *project* 2. *Train* *loss* 3. *Train* MAE 4. *Validation* *loss* 5. *Validation* MAE 6. *Timestamp* 7. Nama *trainer* |

Tabel 3.11 Contoh data flat file ruang warna

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| experiment\_real\_01 | 0.007674 | 0.053751 | 0.041996 | 0.109944 | 2019-06-07\_20:09:14.260621 | adn |
| experiment\_real\_01 | 0.004827 | 0.045996 | 0.035477 | 0.098583 | 2019-06-07\_20:10:44.291938 | adn |

## 3.6 Implementasi Program

Program dijalankan pada *environment* Python dengan *library* utama yang terdiri dari TensorFlow, Keras, OpenCV, dan Flask. TensorFlow, Keras, dan OpenCv memiliki peran utama dalam *preprocessing* data dan fungsionalitas model CNN yang digunakan. Sedangkan Flask digunakan untuk tujuan perancangan antarmuka khususnya sebagai penghubung antara file html sebagai *front-end* dengan jalannya algoritma *back-end* yang dituliskan dengan Python. Dalam penelitian ini, program dijalankan dengan *hardware* dengan spesifikasi berikut.

1. Processor: Intel® Core™ i5-4300U CPU
2. GPU: NVIDIA GeForce GT 720M
3. Memory: 8 gigabyte RAM

## 3.7 Desain Antarmuka



Gambar 3.18 Halaman “Latih Model” bagian pembuatan project atau buka project

Tahapan pembuatan antarmuka dilakukan pada tahap akhir karena tujuan dari antarmuka untuk mempermudah pengamatan hasil dari penggunaan model CNN dengan data masukan ruang warna yang berbeda. Antarmuka dirancang berbasis *website*. Untuk memungkinkan antarmuka dengan *front-end* berbasis website dan *back-end* yang menjalankan program Python maka *framework* Flask digunakan untuk perancangan antarmuka. Antarmuka yang dirancang memiliki tujuan fungsional yang harus dicapai yaitu fungsi latih, fungsi estimasi kadar pigmen fotosintesis, serta fungsi lihat data nilai akurasi latih. Fungsi latih dicapai dengan komponen antarmuka pada halaman “Latih Model”yang ditunjukan pada Gambar 3.18, Gambar 3.19, dan Gambar 3.20. Antarmuka ini mempermudah *user* menganalisa hasil latih model CNN yang diberikan model ruang warna berbeda sebagai data input. Pengguna dapat memilih jenis ruang warna dan memutuskan untuk menjalankan operasi penghilangan refleksi cahaya putih pada citra daun. Kemudian terdapat tampilan yang menunjukan hasil latih dengan informasi nilai MAE dari model CNN.



Gambar 3.19 Halaman “Latih Model” bagian pengaturan proses latih model



Gambar 3.20 Halaman “Latih Model” bagian hasil latih model

Fungsi estimasi dicapai dengan komponen pada halaman “Estimasi” dalam bentuk tombol untuk mengunggah data citra dan kanvas yang menampilkan hasil unggahan ketika proses unggah berhasil dilakukan. Kemudian, tombol estimasi dapat diakses ketika proses pengunggahan berhasil. Selain itu, fungsi estimasi kadar pigmen juga mencakup tampilan informasi hasil dari estimasi. Tampilan informasi estimasi kadar pigmen terdiri dari informasi estimasi kadar pigmen klorofil, karotenoid, dan antosianin dari masing-masing kombinasi ruang warna yang digunakan sebagai data masukan CNN. Fungsi estimasi terdapat pada halaman “Estimasi” yang ditunjukan dengan Gambar 3.21, Gambar 3.22, Gambar 3.23 dan Gambar 3.24.



Gambar 3.21 Halaman “Estimasi” bagian pengunggahan citra daun dan hasil estimasi



Gambar 3.22 Halaman “Estimasi” bagian rubah model estimasi ruang warna



Gambar 3.23 Halaman “Estimasi” bagian pilihan project yang menyediakan model



Gambar 3.24 Halaman “Estimasi” bagian pilihan model

Untuk fungsi lihat dan hapus *record* data latih berada pada halaman “*Record”* yang ditunjukan dengan Gambar 3.25 dan Gambar 2.26. *User* dapat melihat rekap data yang disajikan secara informatif. Untuk melihat rekap data, *user* memilih *project* terlebih dahulu dengan memencet tombol *“Pilih”* yang sejajar dengan nama *project* yang dipilih. Setelah *project* dipilih akan ditampilkan rekap data nilai akurasi hasil latih dari tiap ruang warna dan *preprocessing* beserta informasi grafik.



Gambar 3.25 Halaman “Record” bagian pilihan project



Gambar 3.26 Halaman “Record” bagian rekap data

## 3.10 Pengujian Akurasi CNN P3Net Terhadap Ruang Warna

Pengamatan dari hasil percobaan dilakukan melalui pengajuan performa model-model CNN yang dibedakan dengan jenis data latih. Perbedaan data latih merupakan perbedaan kombinasi *channel* ruang warna dari citra daun. Pengamatan hasil akurasi dihitung berdasarkan nilai MAE dari hasil estimasi model CNN terhadap data validasi. Nilai MAE yang diharapkan dari hasil latih model CNN pada penelitian ini yaitu < 0.1. Dari hasil pengujian, dapat ditemukan kombinasi ruang warna terbaik untuk mengatasi fenomena *color constancy* dan bagaimana dampak *preprocessing* refleksi cahaya putih pada permukaan daun terhadap akurasi model CNN P3Net.Tabel 3.12 menunjukan aspek penilaian dalam pengujian model CNN.

Tabel 3. 12 Aspek pengujian model CNN

|  |  |  |  |
| --- | --- | --- | --- |
| Ruang warna | *Preprocessing* refleksi cahaya putih | Rata-rata MAE latih | Rata-rata MAE validasi |
| RGB | Tidak |  |  |
| HSV | Tidak |  |  |
| LAB | Tidak |  |  |
| YCbCr | Tidak |  |  |
| RGB + HSV | Tidak |  |  |
| RGB + LAB | Tidak |  |  |
| RGB + YCbCr | Tidak |  |  |
| HSV + LAB | Tidak |  |  |
| HSV + YCbCr | Tidak |  |  |
| LAB + YCbCr | Tidak |  |  |
| RGB | Ya |  |  |
| HSV | Ya |  |  |
| LAB | Ya |  |  |
| YCbCr | Ya |  |  |
| RGB + HSV | Ya |  |  |
| RGB + LAB | Ya |  |  |
| RGB + YCbCr | Ya |  |  |
| HSV + LAB | Ya |  |  |
| HSV + YCbCr | Ya |  |  |
| LAB + YCbCr | Ya |  |  |

## 3.11 Pengujian Antarmuka Aplikasi *Web*

Pengujian aplikasi antarmuka dilakukan dengan pengisian angket oleh responden setelah mencoba aplikasi *web* yang dibuat. Aspek-aspek yang diperhatikan dalam pengujian adalah aspek tampilan dan fungsional dari aplikasi *web*. Berikut adalah rincian angket pengujian antarmuka aplikasi *web*. Penentuan responden yang memberikan penilaian pada angket aplikasi *web* dilakukan berdasarkan latar belakang pendidikan atau pekerjaan dari responden yang setidaknya berhubungan dengan *machine learning* atau penelitian pigmen klorofil. Tabel 3.13 menunjukan aspek-aspek pengujian antarmuka aplikasi *web* yang dibuat untuk penelitian yang dilakukan.

Tabel 3.13 Aspek pengujian antarmuka aplikasi *web*

|  |  |
| --- | --- |
| Pertanyaan | Jawaban |
| Tema warna aplikasi web | 1. Sangat baik 2. Baik 3. Cukup 4. Buruk 5. Sangat buruk |
| Tampilan dan pengaturan komponen pada website (tombol, teks, gambar) | 1. Sangat baik 2. Baik 3. Cukup 4. Buruk 5. Sangat buruk |
| Kejelasan informasi pada aplikasi web | 1. Sangat jelas 2. Jelas 3. Cukup 4. Tidak jelas 5. Sangat tidak jelas |
| Alur penggunaan aplikasi web | 1. Sangat Mudah 2. Mudah 3. Cukup 4. Sulit |

Tabel 3.13 Aspek pengujian antarmuka aplikasi *web* (lanjutan)

|  |  |
| --- | --- |
| Alur penggunaan aplikasi web | 1. Sangat sulit |
| Kemudahan proses input | 1. Sangat Mudah 2. Mudah 3. Cukup 4. Sulit 5. Sangat sulit |
| Fungsionalitas aplikasi web | 1. Sangat baik 2. Baik 3. Cukup 4. Buruk 5. Sangat buruk |
| Kemudahan penggunaan fitur aplikasi web | 1. Sangat Mudah 2. Mudah 3. Cukup 4. Sulit 5. Sangat sulit |
| Saran dan masukan |  |

# BAB IV

**Hasil dan Pembahasan**

## 4.1 Ruang warna tunggal dengan akurasi terbaik tanpa proses *inpaint*

Hasil eksperimen yang dilakukan dengan melakukan proses latih pada model-model yang menggunakan ruang warna tunggal tanpa proses *inpaint* sebagai data masukan dicatat. Kemudian ruang warna tunggal tanpa proses *inpaint* diurutkan berdasarkan rata-rata nilai MAE validasi dari yang terkecil sampai yang terbesar seperti yang ditunjukan pada Gambar 4.1. Ruang warna yang dibandingkan di dalam tabel nilai MAE ruang warna adalah ruang warna yang tidak menimbulkan gejala *overfitting*. Tabel 4.1 menunjukan urutan hasil eksperimen latih terhadap ruang warna tunggal tanpa proses *inpaint*.

Gambar 4.1 Grafik urutan nilai akurasi model ruang warna tunggal tanpa inpaint

Tabel 4.1 Urutan nilai akurasi model ruang warna tunggal tanpa inpaint

|  |  |  |  |
| --- | --- | --- | --- |
| No | Ruang warna | Rata-rata nilai MAE latih | Rata-rata nilai MAE Validasi |
| 1 | RGB | 0.05031 | 0.10115 |
| 2 | LAB | 0.05738 | 0.10221 |
| 3 | YCbCr | 0.06376 | 0.10603 |

Dari data yang didapat ditunjukan bahwa dua ruang warna tunggal tanpa proses *inpaint* yang direkomendasikan untuk data masukan dalam penggunaan CNN sebagai metode estimasi kadar pigmen adalah ruang warna RGB dan LAB. Nilai rata-rata MAE validasi dari ruang warna RGB adalah 0.10115, sedangkan untuk ruang warna LAB adalah 0.10221.

## 4.2 Ruang warna kombinasi dengan akurasi terbaik tanpa proses *inpaint*

Hasil eksperimen yang dilakukan dengan melakukan proses latih pada model-model yang menggunakan ruang warna kombinasi tanpa proses *inpaint* sebagai data masukan dicatat. Kemudian ruang warna kombinasi tanpa proses *inpaint* diurutkan berdasarkan rata-rata nilai MAE validasi dari yang terkecil sampai yang terbesar seperti yang ditunjukan dengan Gambar 4.2. Tabel 4.2 menunjukan urutan hasil eksperimen latih terhadap ruang warna tunggal tanpa *inpaint*. Dari data yang didapat ditunjukan bahwa dua ruang warna yang direkomendasikan untuk data masukan dalam penggunaan CNN sebagai metode estimasi kadar pigmen adalah ruang warna RGB+LAB dan RGB+YCbCr. Nilai rata-rata MAE validasi dari ruang warna RGB+LAB adalah 0.09820, sedangkan untuk ruang warna RGB+YCbCr adalah 0.10089.

Gambar 4.2 Grafik urutan nilai akurasi model ruang warna kombinasi tanpa inpaint

Tabel 4.2 Urutan nilai akurasi model ruang warna kombinasi tanpa inpaint

|  |  |  |  |
| --- | --- | --- | --- |
| No | Ruang warna | Rata-rata nilai MAE latih | Rata-rata nilai MAE Validasi |
| 1 | RGB+LAB | 0.05020 | 0.09820 |
| 2 | RGB+YCbCr | 0.05033 | 0.10089 |
| 3 | LAB+YCbCr | 0.06322 | 0.10171 |

## 4.3 Perbandingan akurasi ruang warna tunggal dan kombinasi tanpa proses *inpaint*

Gambar 4.3 Grafik urutan nilai akurasi model ruang warna tunggal dan kombinasi tanpa inpaint

Jenis ruang warna tanpa menggunakan proses *inpaint* yang direkomendasikan didapat dengan membandingkan secara nilai rata-rata MAE validasi dari ruang warna secara keseluruhan baik ruang warna tunggal maupun ruang warna kombinasi. Data nilai rata-rata MAE ruang warna tunggal dan kombinasi ditunjukan dengan Gambar 4.3 dan Gambar 4.4. Tabel 4.3 menunjukan data ruang warna yang telah diurutkan berdasarkan nilai rata-rata MAE validasi dan modelnya tidak mengalami *overfitting*. Dari eksperimen yang dilakukan 3 ruang warna tanpa proses *inpaint* yang direkomendasikan menjadi data masukan CNN sebagai metode estimasi kadar pigmen daun adalah RGB+LAB, RGB+YCbCr, dan RGB. Tiga ruang warna terbaik yaitu RGB+LAB, RGB+YCbCr, dan RGB memiliki nilai rata-rata MAE validasi 0.09820, 0.10089, dan 0.10115.

Gambar 4.4 Grafik urutan nilai akurasi model ruang warna tunggal dan kombinasi tanpa inpaint (lanjutan)

Tabel 4.3 Urutan nilai akurasi semua model ruang warna tanpa inpaint

|  |  |  |  |
| --- | --- | --- | --- |
| No | Ruang warna | Rata-rata nilai MAE latih | Rata-rata nilai MAE Validasi |
| 1 | RGB+LAB | 0.05020 | 0.09820 |
| 2 | RGB+YCbCr | 0.05033 | 0.10089 |
| 3 | RGB | 0.05031 | 0.10115 |
| 4 | LAB+YCbCr | 0.06322 | 0.10171 |
| 5 | LAB | 0.05738 | 0.10221 |
| 6 | YCbCr | 0.06376 | 0.10603 |

## 4.4 Ruang warna tunggal dengan akurasi terbaik dengan proses *inpaint*

Hasil eksperimen dalam bentuk nilai rata-rata MAE validasi dari ruang warna dengan proses *inpaint* diperoleh dengan melakukan proses latih pada model-model CNN masing-masing ruang warna yang sebelumnya melalui proses *inpaint*. Tabel 4.4 menunjukan urutan nilai rata-rata MAE validasi ruang warna tunggal dengan proses *inpain*t sebagai data masukan pada model CNN dari yang terkecil hingga yang terbesar.

Gambar 4.5 Grafik urutan nilai akurasi model ruang warna tunggal dengan inpaint

Tabel 4.4 Urutan nilai akurasi model ruang warna tunggal dengan inpaint

|  |  |  |  |
| --- | --- | --- | --- |
| No | Ruang warna | Rata-rata nilai MAE latih | Rata-rata nilai MAE Validasi |
| 1 | RGB | 0.05173 | 0.09696 |
| 2 | YCbCr | 0.07835 | 0.10049 |
| 3 | LAB | 0.06606 | 0.10333 |

Nilai rata-rata MAE validasi ruang warna tunggal dengan proses *inpainting* menunjukan 2 ruang warna yang direkomendasikan adalah RGB dan YCbCr dengan nilai rata-rata MAE validasi sebesar 0.09252 dan 0.10049.

## 4.5 Ruang warna kombinasi dengan akurasi terbaik dengan proses *inpaint*

Hasil eksperimen yang dilakukan dengan melakukan proses latih model CNN dengan ruang warna kombinasi yang didapat dari citra yang telah melalui proses *inpainting* ditunjukan pada Gambar 4.6. Sedangkan Tabel 4.5 adalah urutan ruang warna kombinasi dengan proses *inpainting* yang telah diurutkan berdasarkan nilai rata-rata MAE validasi dari yang terkecil hingga yang terbesar dan tidak mengalami *overfitting*.

Gambar 4.6 Grafik urutan nilai akurasi model ruang warna kombinasi dengan inpaint

Tabel 4.5 Urutan nilai akurasi model ruang warna kombinasi dengan inpaint

|  |  |  |  |
| --- | --- | --- | --- |
| No | Ruang warna | Rata-rata nilai MAE latih | Rata-rata nilai MAE Validasi |
| 1 | RGB+YCbCr | 0.05051 | 0.09486 |
| 2 | LAB+YCbCr | 0.06761 | 0.10239 |

Dari hasil eksperimen latih model CNN menggunakan ruang warna kombinasi dengan proses *inpaint*, diperoleh 2 ruang warna yang direkomendasikan yaitu ruang warna RGB+YCbCr dan LAB+YCbCr. Ruang warna kombinasi dengan proses *inpaint* RGB+YCbCr dan LAB+YCbCr memperoleh nilai rata-rata MAE validasi sebanyak 0.09486 dan 0.10239.

## 4.6 Perbandingan akurasi ruang warna tunggal dan kombinasi dengan proses *inpaint*

Jenis ruang warna dengan proses *inpaint* yang direkomendasikan dapat diperoleh dengan mendapatkan ruang warna dengan nilai rata-rata MAE validasi terkecil dari keseluruhan ruang warna baik ruang warna tunggal maupun kombinasi. Gambar 4.7 dan Gambar 4.8 menunjukan nilai rata-rata MAE tiap ruang warna baik untuk latih maupaun validasi sedangkan untuk urutan ruang warna dengan proses *inpaint* yang tidak mengalami *overfitting* ditunjukan dengan Tabel 4.6.

Gambar 4.7 Grafik urutan nilai akurasi model ruang warna tunggal dan kombinasi dengan inpaint

Tabel 4.6 Urutan nilai akurasi model ruang warna tunggal dan kombinasi dengan inpaint

|  |  |  |  |
| --- | --- | --- | --- |
| No | Ruang warna | Rata-rata nilai MAE latih | Rata-rata nilai MAE Validasi |
| 1 | RGB+YCbCr | 0.05051 | 0.09486 |
| 2 | RGB | 0.05173 | 0.09696 |
| 3 | RGB+LAB | 0.04471 | 0.09831 |
| 4 | HSV+LAB | 0.03765 | 0.09861 |
| 5 | YCbCr | 0.07835 | 0.10049 |
| 6 | HSV | 0.03545 | 0.10079 |
| 7 | LAB+YCbCr | 0.06761 | 0.10239 |
| 8 | LAB | 0.06606 | 0.10333 |

Dari hasil eksperimen dapat diketahui bahwa jenis ruang warna dengan proses *inpaint* yang direkomendasikan berdasarkan nilai rata-rata MAE validasinya adalah RGB+YCbCr, RGB, dan RGB+LAB. Nilai rata-rata MAE validasi dari ruang warna RGB+YCbCr adalah 0.09486, ruang warna RGB adalah 0.09696, dan untuk RGB+LAB adalah 0.09831.

## 4.7 Perbandingan antara ruang warna terbaik yang melalui proses *inpaint* dan tidak melalui proses *inpaint*

Hasil eksperimen menunjukan bahwa 3 jenis data masukan model CNN estimasi kadar pigmen dengan nilai rata-rata MAE validasi terbaik adalah ruang warna RGB+LAB tanpa *inpaint* dengan nilai rata-rata MAE validasi 0.08761, RGB dengan *inpaint* dengan nilai rata-rata MAE validasi 0.09252, dan LAB+YCbCr tanpa *inpaint* dengan nilai rata-rata MAE validasi 0.09288. Dari hasil yang didapat, ruang warna yang direkomendasikan yaitu RGB+LAB tanpa *inpaint*, RGB dengan *inpaint*, dan LAB+YCbCr tanpa *inpaint*. Tabel 4.7 menunjukan hasil eksperimen seluruh ruang warna tunggal dan kombinasi baik dengan proses *inpainting* maupun tidak. Tabel 4.7 menunjukan urutan model ruang warna dan jenis *preprocess* yang diurutkan berdasarkan nilai rata-rata MAE validasinya serta tidak mengalami *overfitting*.

Tabel 4.7 Urutan nilai akurasi terbaik semua model ruang warna dan preprocessing inpaint

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Ruang warna | *Inpaint* | Nilai rata-rata MAE latih | Nilai rata-rata MAE validasi |
| 1 | RGB+LAB | no | 0.05020 | 0.08761 |
| 2 | RGB | yes | 0.05173 | 0.09252 |
| 3 | LAB+YCbCr | no | 0.06322 | 0.09288 |
| 4 | RGB+YCbCr | yes | 0.05051 | 0.09486 |
| 5 | RGB | no | 0.05031 | 0.09519 |
| 6 | RGB+YCbCr | no | 0.05033 | 0.09594 |
| 7 | LAB | no | 0.05738 | 0.09737 |
| 8 | YCbCr | yes | 0.06489 | 0.10126 |

## 4.8 Tampilan Antarmuka Aplikasi

1. Halaman *“Home”*

Halaman “*Home*” sebagaimana ditunjukan dengan Gambar 4.9 berisi tombol yang dapat mengarahkan *user*  pada halaman “Latih Model” dan halaman “Estimasi”. Pada Gmabar ditunjukan tombol 1 mengarah pada halaman “Estimasi”, sedangkan tombol 2 mengarah pada halaman “Latih Model”.



Gambar 4.9 Hasil tampilan halaman “Home”

1. Halaman “Latih Model”

Pada halaman “Latih Model”, *user* dapat melakukan proses latih model sesuai dengan ruang warna dan konfigurasi *preprocessing* yang diinginkan. Gambar 4.10, Gambar 4.11, Gambar 4.12, dan Gambar 4.13 menunjukan perubahan tampilan sesuai dengan alur penggunaan dari fungsi latih model. Pada tahap awal ditunjukan dengan Gambar 4.10, *user* dapat memilih untuk membuat *project* baru dengan memencet tombol “*New Project”* yang ditunjukan dengan angka 1 atau memilih untuk menggunakan *project* yang telah dibuat dengan memencet tombol ”*Open Project”* yang ditunjukan dengan angka 2. Jika *user* memilih untuk membuat *project* baru, tampilan akan berubah seperti yang ditunjukan pada Gambar 4.11. Pada Gambar 4.10, *user* harus memasukan nama *project* dan memencet tombol “*Create*” yang ditunjukan dengan angka 1. Pada Gambar 4.12 *user* harus memilih *project* dengan memencet tombol “Pilih” yang ditunjukan dengan angka 1. Kemudian setelah *user* membuat *project* baru atau memilih *project* yang telah dibuat, *user* akan diarahkan pada tampilan dengan konfigurasi latih model yang ditunjukan dengan angka 1 pada Gambar 4.13. Pada tampilan tersebut, *user* diharuskan memilih ruang warna dengan mencentang *checkbox* pada kotak “Pilihan Ruang Warna” dan memilih setelan *preprocessing inpaint* pada *checkbox inpaint.* Kemudian *user* memencet tombol “Latih” yang ditunjukan dengan angka 2 pada Gambar 4.13. Setelah proses latih selesai dan tampilan nilai akurasi dari model yang telah dilatih muncul, *user* dapat menyimpan nilai akurasi dengan memencet tombol “Simpan” yang ditunjukan dengan angka 3 pada Gambar 4.13.



Gambar 4.10 Hasil tampilan halaman “Latih Model” tahap awal



Gambar 4.11 Hasil tampilan halaman “Latih Model” bagian create project



Gambar 4.12 Hasil tampilan halaman “Latih Model” bagian open new project



Gambar 4.13 Hasil tampilan halaman “Latih Model” bagian latih model

1. Halaman estimasi



Gambar 4.14 Hasil tampilan halaman “Estimasi” bagian estimasi



Gambar 4.15 Hasil tampilan halaman “Estimasi” bagian estimasi (lanjutan)

Pada halaman “Estimasi” yang ditunjukan pada Gambar 4.14 dan Gambar 4.15, terdapat elemen yang memungkinkan *user* untuk melakukan estimasi kadar pigmen dari citra daun. *User* dapat merubah *file* model yang akan digunakan pada proses estimasi dengan memencet tombol “*Edit Model”* yang ditunjukan dengan angka 1 pada Gambar 4.14. Setelah *user* yakin terhadap *file* model yang digunakan, *user* dapat mengunggah citra daun yang akan diestimasi kadar pigmennya dengan memencet tombol “*Choose File”* yang ditunjukan dengan angka 2 pada Gambar 4.14. Setelah citra daun selesai diunggah, *user* dapat memencet tombol “Estimasi” yang ditunjukan dengan angka 3 pada Gambar 4.14 untuk memulai proses estimasi. Ketika proses estimasi kadar pigmen selesai, akan muncul hasil dari kadar pigmen fotosintesis dari tiap kombinasi ruang warna dan *preprocessing inpaint* seperti yang ditunjukan pada Gambar 4.15.

1. Halaman *record*

Pada halaman *record, user* dapat melihat rekap data dari nilai akurasi latih model yang sudah dilatih pada masing-masing *project*. Halaman *record* ditunjukan dengan Gambar 4.16, Gambar 4.17, dan Gambar 4.18. Pada tahap awal, *user* memilih *project* yang tersedia dengan memencet tombol “Pilih” disamping *project* yang akan dipilih, tombol tersebut ditunjukan dengan angka 1 pada Gambar 4.16. Setelah tombol “Pilih” dipencet, tampilan akan berubah menjadi kumpulan tabel dan grafik rekap data sebagaimana ditunjukan dengan Gambar 4.17 dan Gambar 4.18. Pada tampilan rekap data, *user* dapat mengunduh data rekap data dalam format CSV dengan memencet tombol “*Download”* yang ditunjukan dengan angka 1 pada Gambar 4.17. *User* dapat menghapus data nilai akurasi dengan memencet tombol “Hapus” yang ditunjukan dengan angka 3 pada Gambar 4.17. Jika *user* ingin melihat rekap data *project* berbeda, *user* dapat memencet tombol “Kembali” yang ditunjukan dengan angka 2 pada Gambar 4.17 untuk kembali pada tampilan daftar *project* yang tersedia.



Gambar 4.16 Hasil tampilan halaman “Record” bagian pilih project



Gambar 4.17 Hasil tampilan halaman “Record” bagian rekap data



Gambar 4.18 Hasil tampilan halaman “Record” bagian rekap data (lanjutan)

## 4.9 Penilaian Antarmuka Aplikasi Web

Penilaian antarmuka aplikasi *web* dilakukan dengan penilaian oleh 3 responden yang memiliki latar belakang pekerjaan yang berhubungan dengan aplikasi yaitu CNN dan penelitian pigmen fotosintess daun. Responden pertama yaitu Bapak Marcelinus A.S. Adhiwibawa selaku asisten peneliti di MRCPP (*Ma Chung Research Center for Photosynthetic Pigments*), yang kedua adalah Dr. Tatas Brotosudarmo, Ph.d selaku peneliti utama di MRCPP, dan yang ketiga adalah Ibu Kestrilia Rega Prilianti, M.Si selaku dosen program studi teknik informatika di Universitas Ma Chung. Berikut merupakan hasil penilaian antarmuka aplikasi *web* yang diperoleh dari kuesioner sebagaimana ditunjukan dengan Lampiran 1.

1. Tema warna pada *web*

Gambar 4.19 Grafik hasil penilaian tema warna web

Dari Gambar 4.19 dapat diketahui bahwa 2 responden menilai tema warna pada *web* termasuk baik, sedangkan terdapat 1 responden yang menilai bahwa tema warna pada *web* termasuk cukup.

1. Tampilan dan pengaturan komponen pada *website* (tombol, teks, gambar)

Dari Gambar 4.20 dapat diketahui bahwa 2 responden menilai tampilan dan pengaturan komponen pada *website* termasuk baik, sedangkan terdapat 1 responden yang menilai bahwa tampilan dan pengaturan komponen pada *website* termasuk cukup.

Gambar 4.20 Grafik hasil penilaian tampilan dan pengaturan komponen pada web

1. Kejelasan informasi yang ada pada *website*

Dari Gambar 4.21 dapat diketahui bahwa 2 responden menilai kejelasan informasi yang ada pada *website* termasuk cukup, sedangkan terdapat 1 responden yang menilai bahwa kejelasan informasi yang ada pada *website* termasuk tidak jelas.

Gambar 4.21 Grafik hasil penilaian kejelasan informasi pada website

1. Alur penggunaan aplikasi *web*

Dari Gambar 4.22 dapat diketahui bahwa 1 responden menilai alur penggunaan aplikasi *web* termasuk mudah, 1 responden menilai alur penggunaan aplikasi *web* termasuk cukup, dan 1 responden menilai alur penggunaan aplikasi *web* termasuk sulit.

Gambar 4.22 Grafik hasil penilaian alur penggunaan aplikasi web

1. Kemudahan proses *input*

Dari gambar 4.23 dapat diketahui bahwa 1 responden menilai kemudahan proses *input* termasuk mudah, 1 responden menilai kemudahan proses *input* termasuk cukup, dan 1 responden menilai kemudahan proses *input* termasuk sulit.

Gambar 4.23 Grafik hasil penilaian kemudahan proses input

1. Fungsionalitas aplikasi *web*

Dari Gambar 4.24 dapat diketahui bahwa 2 responden menilai fungsionalitas aplikasi *web* termasuk baik, sedangkan terdapat 1 responden yang menilai bahwa fungsionalitas aplikasi *web* termasuk cukup.

Gambar 4.24 Grafik hasil penilaian fungsionalitas aplikasi web

1. Kemudahan penggunaan fitur aplikasi *web*

Gambar 4.25 Grafik hasil penilaian kejelasan informasi pada website

Dari Gambar 4.25 dapat diketahui bahwa 1 responden menilai kemudahan penggunaan fitur aplikasi *web* termasuk baik, sedangkan terdapat 2 responden yang menilai bahwa kemudahan penggunaan fitur aplikasi *web* termasuk cukup.

1. Saran dan masukan untuk aplikasi *web*

Saran dan masukan dari 3 responden dari angket meliputi aspek fungsionalitas serta tampilan dari aplikasi web. Dari segi fungsionalitas saran yang diberikan adalah meningkatkan hasil prediksi kadar pigmen sehingga lebih akurat. Dari segi tampilan saran yang diberikan adalah perbaikan tata letak elemen pada aplikasi *web* sehingga lebih *user friendly*, kemudian perlunya diberikan keterangan yang lebih deskriptif.

# BAB V

**Simpulan dan Saran**

## 5.1 Simpulan

Eksperimen yang dilakukan dengan melakukan proses latih model CNN menggunakan jenis ruang warna dan setelan *preprocessing* berbeda menunjukan perbedaan nilai akurasi latih dan validasi. Dengan mengurutkan jenis ruang warna dan setelan *preprocessing* berdasarkan nilai rata-rata MAE validasi, jenis ruang warna yang direkomendasikan untuk data masukan model CNN P3Net estimasi kadar pigmen dapat ditentukan. Untuk data masukan tanpa proses *inpaint* yang direkomendasikan adalah ruang warna RGB+LAB, RGB+YCbCr, dan RGB. Namun untuk data masukan tanpa proses *inpaint* dengan ruang warna tunggal, ruang warna yang direkomendasikan adalah RGB dan LAB. Sedangkan untuk data masukan tanpa proses *inpaint* dengan ruang warna kombinasi, yang direkomendasikan adalah RGB+LAB dan RGB+YCbCr.

Hasil eksperimen pelatihan model CNN P3Net menggunakan proses *inpaint* pada data citra menghasilkan kesimpulan bahwa dari keseluruhan ruang warna, yang direkomendasikan adalah RGB+YCbCr, RGB, dan RGB+LAB. Ruang warna tunggal yang direkomendasikan untuk data masukan dengan proses *inpaint* adalah RGB dan YCbCr. Sedangkan ruang warna kombinasi yang direkomendasikan untuk data masukan dengan proses *inpaint* adalah RGB+YCbCr dan LAB+YCbCr. Untuk perbandingan seluruh ruang warna dan setelan *preprocessing inpaint,* ruang warna yang direkomendasikan adalah RGB+LAB tanpa *inpaint*, RGB dengan *inpaint*, dan LAB+YCbCr tanpa inpaint. Hasil eksperimen menunjukan bahwa jenis ruang warna dan setelan *preprocessing inpaint* mempengaruhi akurasi dari model CNN P3Net.

Untuk antarmuka aplikasi *web,* penilaian terhadap aspek tampilan dinilai cukup dan tidak mendapatkan penilaian buruk. Sedangkan dari segi fungsionalitas ada beberapa penilaian yang dibawah cukup yaitu dari kejelasan informasi, alur penggunaan aplikasi *web*, dan kemudahan proses *input*. Untuk aspek fungsionalitas

yang terdiri dari fungsionalitas aplikasi *web,* dan kemudahan penggunaan fitur aplikasi *web* mendapatkan penilaian cukup dan di atas cukup.

## 5.2 Saran

Berdasarkan penelitian ini dan pengujian yang dilakukan, berikut merupakan saran-saran yang dapat dibuat dan dikembangkan di penelitian selanjutnya:

1. Mengembangkan metode *preprocessing* *inpaint* yang digunakan agar dapat lebih optimal dalam memperbaiki warna mengkilat dari citra daun.
2. Memperkecil jarak antar nilai MAE latih dan MAE validasi sehingga *overfitting* tidak terjadi dengan mencoba dan membandingkan berbagai konfigurasi model CNN.

# DAFTAR PUSTAKA

Baker, N.R. 2007. *Advances in Photosynthesis Volume 9*. Dordrecht: Springer.

Bertalmio, M., A. L. Bertozzi, G. Sapiro. 2001. *Navier Stokes, Fluid Dynamics, and Image and Video Inpainting*.

Croft, H.J, J.M Chen. 2017. *Leaf Pigment Content*. Toronto: University of Toronto

Feret, J.B., A.A. Gitelson, S.D. Noble, S. Jacquemoud. 2017. *PROSPECT-D: Towards Modelling Leaf Optical Properties Through a Complete Lifecycle*. *Remote Sensing of Environment.* Amsterdam: Elsevier.

Foster, H. David. 2010. *Color Constancy*. *Vision Research Vol 5 Issue 37.* Amsterdam: Elsevier.

Gary, Bradski, Adrian Kaehler. 2008. *Learning OpenCV*. Sebastopol: O’Reilly Media, Inc.

Geusebroek, Jan-Mark, Rein van den Boomgaard, Arnold W.M Smeulders, Theo Gevers. 2003. *Color Constancy From Physical Principles*. *Pattern Recognition Letters 24*. Amsterdam: John Elsevier.

Google. (12 Desember 2018). *Tensorflow Guide*. Diperoleh 20 Februari 2019, dari <https://www.tensorflow.org/guide/>

Guttag, John V, 2013, *Introduction to Computation and Programming Using Python,* Massachuseetts: MIT Press.

Hasibuan, Eliwati. 2015. Pengenalan Spektrofotmetri Pada Mahasiswa yang Melakukan Penelitian di Laboratorium Terpadu Fakultas Kedokteran USU. Medan: Fakultas Kedokteran Universitas Sumatera Utara.

Johnson, P. Mathew. 2016. *Photosynthesis*. *Essay in Biochemistry 255-273.* Sheffield: Portland Press.

Kamilaris, Andreas, Francesc Xavier Prenafeta Boldu. 2018. *Deep Learning in Agriculture: A Survey*. *Computers and Electronics in Agriculture.*

Prilianti, K. R., Onggara, I. C., & Setiawan, H. (2018). Implementasi *Convolutional Neural Network* untuk sistem Prediksi Pigmen Fotosintesis pada Tanaman Secara Real Time. *Jurnal Teknik Informatika dan Sistem Informasi*, *4*(2).

Prilianti, K.R., T.H.P Brotosudarmo, S. Anam, A. Suryanto. *Performance Comparison of the Convolutional Neural Network Optimizer for Photosynthetic Pigments Prediction on Plant Digital Image*. *Proceeding Symposium on Biomathematics.*

Speer, B. R. (1997). *Photosynthetic Pigments.* Diperoleh20 Januari 2019, dari http://www.ucmp.berkeley.edu/glossary/gloss3/pigments.html

Vesali, Farshad, Mahmoud Omid, Amy Kaleita, Hossein Mobli. 2015. *Development of an android app to estimate chlorophyll content of corn leaves based on contact imaging*. Amsterdam: Elsevier.

Zivkovic, Nikola. (26 Februari 2018). *Introduction to Convolutional Neural Networks*. Diperoleh 20 Januari 2019, dari <https://rubikscode.net/2018/02/26/introduction-to-convolutional-neural-networks/>

# Lampiran 1







