

Implementasi *Convolutional Neural Network* untuk Sistem Prediksi Pigmen Fotosintesis pada Tanaman Secara *Real Time*

<http://dx.doi.org/10.28932/jutisi.v4i2.812>

Kestrilia Rega P¹, Ivan Christianto O², Hendry Setiawan³

Teknik Informatika, Universitas Ma Chung
Villa Puncak Tidar N-01, Malang-Jawa Timur

¹kestrilia.rega@machung.ac.id

²311410009@student.machung.ac.id

³hendry.setiawan@machung.ac.id

Abstract — It is common that evaluation on plant health is done by conducting measurement on photosynthetic pigments. Analysis of the presence or absence of some particular pigments could reveal any information about plant responses to the environment or climate changes. This is due to the fact that relative pigment concentrations are influenced by environmental factors such as light and nutrient availability. In this research, a non-destructive and rapid method was developed to identify the existence of photosynthetic pigments in plant leaf i.e. chlorophyll, carotenoid, and anthocyanin. The method used leaf's RGB digital image as the color representation of the pigments contained in the plant being evaluated. The intelligence agent which is responsible to learn the data and provide information about the pigments was developed based on convolutional neural network (CNN) model. This model was chosen due to its capability to receive a digital image and automatically search for the best feature to learn it. Therefore, plant evaluation could run in real time. The result of the experiment reveals that CNN model could learn the color-pigment relationship very well. The best architecture is *ShallowNet* using Adam optimizer, batch size 30 and trained with 15 epoch. The MSE of the pigments prediction reaches 0.0055 (actual data range -0.2 up to 2.2) for training and 0.029 for testing.

Keywords—Convolutional Neural Network (CNN), photosynthetic pigment, digital image, plant health.

I. PENDAHULUAN

Pada umumnya kondisi tanaman dipantau melalui keragaman warnanya. Warna yang tampak dari tanaman dapat memberikan beragam indikasi seperti kecukupan nitrogen dan air, potensi serangan hama dan penyakit, usia dan kematangan buah serta perubahan kondisi lingkungan. Hal tersebut menjadi rasional karena pada dasarnya warna dari tanaman sangat ditentukan oleh karakteristik spektral pigmen yang merupakan komponen utama dalam proses penyerapan energi dari sinar matahari untuk proses fotosintesis (sehingga disebut juga pigmen fotosintesis). Karakteristik spektral pigmen terkait dengan kemampuan

menyerap foton pada suatu panjang gelombang tertentu dan memantulkan foton pada panjang gelombang lainnya. Karakteristik inilah yang hingga kini dijadikan dasar oleh sebagian besar peneliti dalam mengembangkan berbagai perangkat sensor dan metode prediksi untuk melakukan evaluasi terhadap kandungan pigmen [1]. Klorofil, Karotenoid dan Antosianin merupakan pigmen fotosintesis yang paling mudah dilihat secara visual melalui keragaman warna pada daun tanaman. Komposisinya sangat dipengaruhi oleh jumlah cahaya, oksigen, nutrisi, air dan suhu lingkungan dimana tanaman tersebut tumbuh [2].

Berdasarkan prosesnya, analisis pigmen secara garis besar terbagi menjadi dua teknik yaitu destruktif dan non-destruktif. Teknik destruktif yang umum digunakan adalah Kromatografi Cair Kinerja Tinggi (KCKT). Namun, selain membutuhkan keahlian dan tempat khusus (laboratorium), metode tersebut membutuhkan banyak tahapan dalam penanganan sampelnya sehingga tidak dapat memproduksi data secara *real time*. Oleh karena itu metode destruktif tidak efisien untuk digunakan sehari-hari. Metode non-destruktif yang hingga kini masih populer digunakan sebagai alternatif metode destruktif adalah spektroskopi optik [3] dan pencitraan digital [4].

Perangkat spektroskopi optik dapat menghasilkan data yang lebih akurat dibandingkan dengan perangkat pencitraan digital yang pada umumnya dapat menggunakan kamera digital berwarna dengan *output* citra RGB. Namun, harga perangkat spektroskopi jauh lebih mahal daripada kamera digital sehingga kurang cocok untuk pengembangan perangkat portabel berbiaya rendah. Oleh karena itu, pada penelitian ini dikaji kinerja dari citra RGB untuk dapat digunakan sebagai *raw data* pada sistem prediksi pigmen fotosintesis. Dalam upaya menghadirkan sistem yang dapat secara *real time* melakukan analisis pada citra RGB, maka dikembangkan pula model berbasis *soft computing* yang dapat secara otomatis memilih dan menemukan fitur utama pada citra digital dan menghubungkannya dengan kandungan pigmen yang ada.

Model tersebut adalah *Convolutional Neural Network* (CNN).

Model CNN telah sangat populer digunakan untuk klasifikasi citra digital yang utamanya memanfaatkan bentuk sebagai fitur pembeda utama [5,6]. Tidak banyak penelitian membahas pengembangan model CNN dengan fitur utama warna. Namun, beberapa peneliti telah membuktikan bahwa CNN juga dapat digunakan untuk melakukan analisis objek berdasarkan warnanya [7,8]. Beberapa penelitian CNN pada citra digital tanaman antara lain adalah klasifikasi spesies [9,10] dan deteksi penyakit [11]. Hingga kini belum ditemukan penelitian yang mengkaji implementasi CNN untuk prediksi pigmen fotosintesis sebagaimana yang dilakukan pada penelitian ini.

II. TINJAUAN PUSTAKA

A. Pigmen pada Tanaman dan Sifatnya

Pigmen pada daun tanaman merupakan senyawa kimia dengan ikatan rangkap terkonjugasi (kromofor) yang mempunyai kemampuan untuk menyerap dan memantulkan cahaya pada panjang gelombang tertentu. Sifat inilah yang memungkinkan pigmen menyerap energi matahari dan juga memiliki warna tertentu. Oleh karena itu, pigmen mempunyai peran yang sangat penting pada proses metabolisme tanaman seperti contohnya pada proses fotosintesis yang menggunakan energi foton untuk mengubah air dan karbondioksida menjadi makanan (karbohidrat). Sebuah tanaman pada umumnya mempunyai lebih dari satu jenis pigmen dan secara umum pigmen-pigmen tersebut terbagi menjadi empat kelompok utama, yaitu porifirin, karotenoid, flavonoid dan betalain [12].

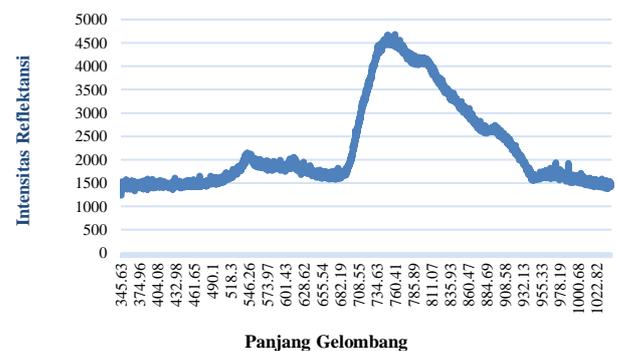
Tiap kelompok pigmen mempunyai karakteristik masing-masing dalam penyerapan cahaya yaitu akan menyerap cahaya pada panjang gelombang tertentu dan memantulkan cahaya pada panjang gelombang lainnya [13]. Secara umum pigmen pada tanaman akan menyerap cahaya tampak (400-700 nm) dan infra merah gelombang pendek (700-1100 nm). Dari sekian banyak pigmen yang ada pada tanaman, kebanyakan peneliti hanya menggunakan Klorofil, Karotenoid dan Antosianin sebagai indikator pemantauan kondisi tanaman karena perannya yang dominan pada proses fotosintesis [14].

Klorofil sangat kuat dalam menyerap cahaya merah pada panjang gelombang 650-700 nm, biru pada panjang gelombang 400-500 nm dan memantulkan cahaya hijau pada panjang gelombang 500-650 nm. Oleh karena itu Klorofil akan memberikan tampilan visual warna hijau pada tanaman. Karotenoid pada umumnya sangat kuat menyerap cahaya biru pada panjang gelombang 400-500 nm dan memantulkan cahaya kuning-merah (500-760 nm). Hal ini yang menyebabkan Karotenoid akan memberikan tampilan visual warna kuning sampai dengan kemerahan pada tanaman. Sedangkan serapan maksimum Antosianin terjadi pada cahaya biru-hijau di panjang gelombang 450-550 nm

dan memantulkan sebagian besar cahaya merah pada panjang gelombang di sekitar 700 nm. Antosianin mempunyai warna visual merah kecoklatan hingga ungu.

B. Perhitungan Kandungan Pigmen dengan Metode Spektroskopi Non-Destruktif

Pada penelitian ini, data target untuk proses pelatihan arsitektur CNN dihasilkan dengan menggunakan metode spektroskopi non-destruktif. Prinsip kerja dari metode tersebut adalah memancarkan cahaya pada objek yang dianalisis pada rentang panjang gelombang tertentu untuk kemudian mencatat banyaknya cahaya yang dipantulkan oleh objek tersebut. Jika dibuat dalam bentuk grafik, maka data yang direkam oleh perangkat spektroskopi (spektrometer) akan tampak seperti Gambar 1. Objek yang dianalisis pada gambar tersebut adalah daun berwarna hijau yang direkam intensitas cahaya yang dipantulkan pada panjang gelombang 345 s.d. 1024 nm. Jumlah kandungan klorofil, Karotenoid dan Antosianin pada daun tanaman yang diambil citra digitalnya dapat dihitung melalui data tersebut dengan mengacu pada protokol yang dibuat oleh Gitelson [15].



Gambar 1. Contoh grafik data yang direkam oleh spektrometer

Persamaan 1 dan 2 adalah rumus untuk menghitung kandungan Klorofil (*Chlorophyll Reflectance Index*). Persamaan 3 dan 4 adalah rumus untuk menghitung kandungan Karotenoid (*Carotenoid Reflectance Index*) dan Persamaan 5 adalah rumus untuk menghitung kandungan Antosianin (*Anthocyanin Reflectance Index*). Nilai R adalah intensitas cahaya pada panjang gelombang tertentu yang dipantulkan oleh daun tanaman. Misalkan R_{510} , variabel tersebut akan memuat intensitas cahaya dengan panjang gelombang 510 nm yang dipantulkan oleh daun tanaman. Kelima nilai yang merepresentasikan kandungan tiga pigmen fotosintesis tersebutlah yang nantinya akan diprediksi oleh CNN.

$$(Chl)RI_{green} = \left[\frac{R_{750-800} - R_{430-470}}{R_{520-580} - R_{440-480}} \right] - 1 \quad (1)$$

$$(Chl)RI_{red\ edge} = \left[\frac{R_{750-800} - R_{430-470}}{R_{659-740} - R_{440-480}} \right] - 1 \quad (2)$$

$$CRI_{green} = [(R_{510})^{-1} - (R_{550-570})^{-1}] * R_{750-800} \quad (3)$$

$$CRI_{red\ edge} = [(R_{510})^{-1} - (R_{700-710})^{-1}] * R_{750-800} \quad (4)$$

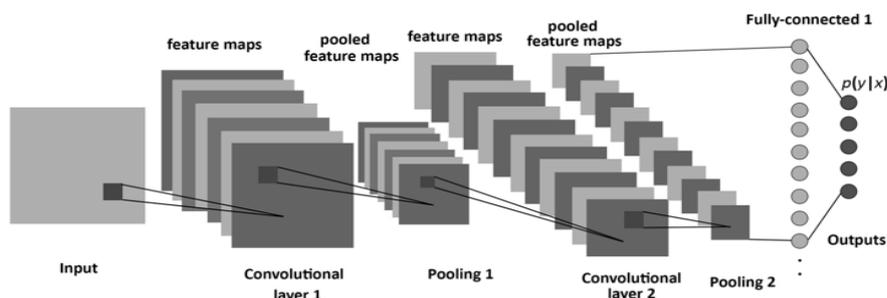
$$ARI = [(R_{500-570})^{-1} - (R_{700-710})^{-1}] * R_{750-800} \quad (5)$$

C. Struktur Dasar Convolutional Neural Network (CNN)

Model CNN merupakan pengembangan dari *Multilayer Perceptron* (MLP) pada Jaringan Syaraf Tiruan (JST) yang didesain untuk mengolah data multi dimensi seperti citra digital berwarna [16]. Model CNN memodifikasi arsitektur JST sedemikian sehingga dapat langsung memproses citra digital dengan konsep-konsep dasar pengolahan citra digital. Modifikasi tersebut dilakukan utamanya melalui penambahan *convolution layer* yang memproses citra digital dengan operasi konvolusi. Matriks filter (biasa disebut juga kernel) yang digunakan pada operasi konvolusi terus diperbaharui sepanjang proses pelatihan. Hal inilah yang membuat CNN dapat secara otomatis memilih fitur yang paling sesuai untuk citra digital yang dikelolanya. Dengan menggunakan JST biasa, peneliti harus sangat cermat dalam memilih fitur citra digital yang akan dijadikan *input*. Dengan CNN proses tersebut adalah bagian dari pelatihan. Pada lapisan akhir CNN yang disebut *fully-connected layer*, proses yang dijalankan tidak jauh berbeda dengan JST pada umumnya [17].

Gambar 2 mengilustrasikan arsitektur dasar dari CNN. *Layer input* berisi citra digital dua dimensi atau lebih (seperti format RGB, HSI, CMYK dan lain sebagainya). *Layer konvolusi* berisi kumpulan filter yang biasa digunakan pada proses *spatial filtering* citra digital. Jika *input* adalah citra digital dengan l dimensi (misal jika menggunakan ruang warna RGB maka disebut 3 dimensi), maka operasi konvolusi dilakukan berdasarkan Persamaan 6 yang pada prinsipnya adalah perkalian produk dari citra digital *input* (f) dengan filter-nya. Nilai x dan y adalah posisi baris dan kolom pada matriks intensitas citra digital *input*, c_z adalah filter berukuran $(c \times c)$ dan hasilnya adalah citra digital g . Filter-filter pada *layer* ini mempunyai kemampuan belajar untuk mengekstraksi ciri lokal yang paling sesuai dari citra digital *input*.

$$g(x, y, z) = \sum_{i=0}^{c-1} \sum_{j=0}^{c-1} \sum_{k=1}^l f(x+i, y+j, k) \cdot c_z(i, j, k) \quad (6)$$



Gambar 2. Arsitektur dasar CNN [18]

Desain arsitektur CNN dapat diubah-ubah jumlah maupun susunan dari *layer konvolusi* dan *layer pooling*. Pada bagian akhir arsitektur terdapat *fully-connected layer* yang akan menggabungkan semua ciri utama dari *layer* sebelumnya. *Layer* ini terhubung dengan *layer output* yang berisi *node-node* dengan fungsi aktivasi seperti *softmax* sebagaimana Persamaan 7. Nilai tersebut merupakan probabilitas (ternormalisasi) sebuah citra input x_i dengan parameter W masuk pada kelas y_i . Jika CNN digunakan untuk tujuan klasifikasi, maka jumlah *node* pada *layer output* adalah sama dengan jumlah kelas.

$$P(y_i|x_i; W) = \frac{e^{f y_i}}{\sum_j e^{f_j}} \quad (7)$$

Tahapan pembelajaran secara umum mengikuti prosedur berikut:

1. Inisialisasi semua filter dan bobot dengan bilangan random
2. Citra digital untuk data latih masuk kedalam jaringan sebagai *input* dan menjalani tahap propagasi maju (konvolusi, *pooling* sampai dengan *fully connected*) dan mendapatkan probabilitas ternormalisasi untuk setiap kelas (setiap *node* pada *output*)
3. Perhitungan total *error* (misal dengan *Mean Square Error*) dari *layer output*.
4. Propagasi balik untuk menghitung gradien dari *error* dan menerapkan metode optimasi (misal *gradient descent*) untuk meminimalkan *error* dari *output*. Setiap bobot dan matriks filter kemudian disesuaikan secara proporsional dengan mempertimbangkan sumbangannya terhadap total *error*.
5. Langkah 2 s.d. 4 diulang-ulang untuk seluruh data latih hingga kondisi berhenti tercapai.

III. METODE PENELITIAN

A. Persiapan Sampel Daun Tanaman

Sejumlah 391 helai daun tanaman digunakan untuk membangun model CNN. Secara umum sampel terbagi menjadi dua kelompok utama. Kelompok pertama adalah helai daun dari tanaman pucuk merah (*Syzygium oleana*) dan kelompok kedua adalah helai daun dari tanaman sirih (*Piper betle*).

Kelompok pertama akan memberikan koleksi daun yang warnanya bergradasi dari hijau pekat hingga merah, merepresentasikan keberadaan dominan pigmen klorofil dan antosianin. Kelompok kedua akan memberikan koleksi daun yang warnanya bergradasi dari hijau pekat hingga kuning cerah, merepresentasikan keberadaan dominan pigmen klorofil dan karotenoid (lihat Gambar 3).



Gambar 3. Tanaman sampel, (a) Pucuk Merah; (b) Sirih

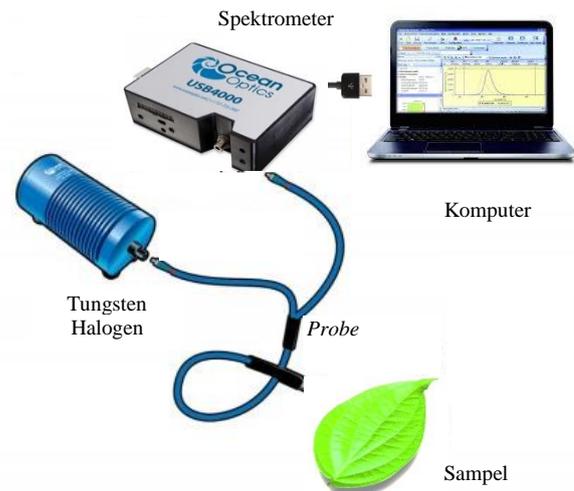
Keseluruhan sampel selanjutnya menjalani 2 jenis akuisisi data. Akuisisi data yang pertama adalah dengan menggunakan kamera digital untuk mendapatkan citra dalam format RGB yang nantinya akan menjadi *input* pada arsitektur CNN. Akuisisi data yang kedua adalah dengan menggunakan spektrometer untuk mendapatkan data intensitas reflektansi tiap sampel pada berbagai panjang gelombang sebagaimana yang ditunjukkan pada Gambar 1. Data tersebut selanjutnya akan diolah dengan Persamaan 1 s.d. 5 untuk mendapatkan kandungan Klorofil, Karotenoid dan Antosianin dari tiap sampel. Data kandungan pigmen akan menjadi *output* pada arsitektur CNN.

B. Akuisisi Citra Digital

Kamera digital yang digunakan pada penelitian ini adalah kamera *smartphone* dengan resolusi 13 MP. Hal ini dilakukan dengan pertimbangan bahwa tujuan akhir dari peta jalan penelitian ini adalah mengembangkan *portable device* yang dapat digunakan secara mudah, cepat dan *real time* untuk analisis pigmen tanaman. Oleh karena itu kajian terhadap kualitas data *input* sangatlah penting. Latar belakang sampel saat pengambilan foto dikendalikan dengan menggunakan kertas putih.

C. Akuisisi Data Kandungan Pigmen

Proses akuisisi data intensitas reflektansi menggunakan spektrometer dilakukan dengan perangkat yang disusun sebagaimana Gambar 4. *Probe* menyalurkan cahaya yang dihasilkan oleh sumber cahaya (*Tungsten Halogen*) menuju ke permukaan sampel. Cahaya yang dipantulkan oleh sampel selanjutnya ditangkap oleh *probe* dikirimkan ke spektrometer untuk kemudian diukur intensitasnya. Hasil pengukuran diolah dan ditampilkan oleh perangkat lunak di komputer.



Gambar 4. Pengaturan perangkat akuisisi intensitas reflektansi

D. Desain Dataset

Tabel I menunjukkan organisasi data yang digunakan pada penelitian ini. Setiap citra digital daun akan berpasangan dengan 5 nilai indeks reflektansi yang dihitung dengan Persamaan 1 s.d. Persamaan 5. Dari keseluruhan data tersebut, 80% digunakan sebagai data pelatihan sedangkan 20% sisanya digunakan sebagai data uji.

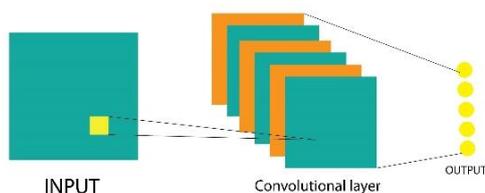
TABEL I
ORGANISASI DATA HASIL EKSPERIMEN

File citra digital		Daun1.jpg	...	Daun391.jpg
Indeks Reflektansi	$(Chl)RI_{green}$	0.54	...	0.98
	$(Chl)RI_{rededge}$	1.32	...	1.54
	CRI_{green}	-0.81	...	0.04
	$CRI_{rededge}$	0.48	...	0.01
	ARI	0.72	...	0.01

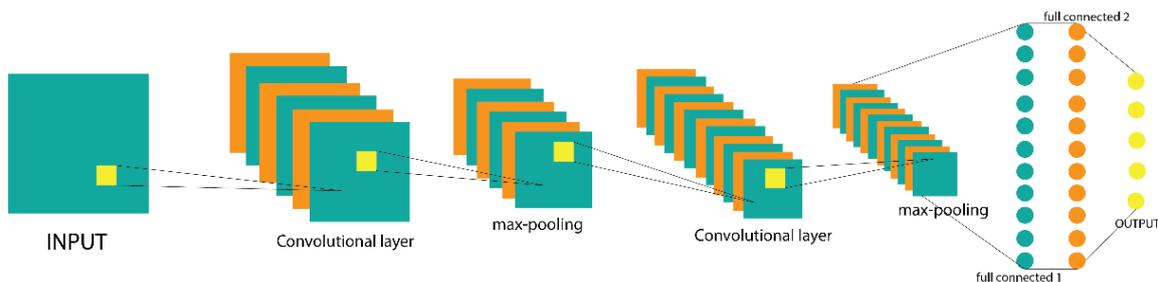
E. Desain Arsitektur CNN

Tiga jenis arsitektur CNN yaitu *ShallowNet*, *LeNet*, dan *AlexNet* digunakan dan dikaji kinerjanya pada penelitian ini. Citra digital *input* menjalani proses *resize* terlebih dahulu menjadi ukuran yang sesuai dengan standar masing-masing arsitektur. Proses tersebut dilakukan dengan menggunakan *library OpenCV*. Pada arsitektur *ShallowNet* dan *LeNet*, citra *input* diubah menjadi berukuran 32x32 piksel, sedangkan pada arsitektur *AlexNet* citra *input* diubah menjadi berukuran 120x120 piksel. Pemilihan data pelatihan dan data uji dilakukan secara *random* oleh sistem. Berikut adalah desain untuk masing-masing tipe arsitektur:

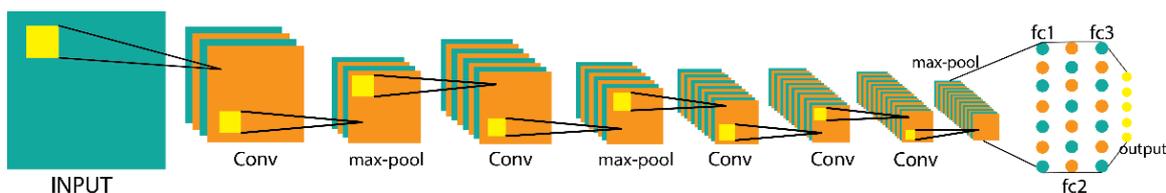
1) *ShallowNet*: Gambar 5 menunjukkan ilustrasi desain arsitektur yang digunakan. Terdapat 1 *convolution layer* yang berisi 32 filter berukuran 3x3.



Gambar 5. Ilustrasi desain arsitektur *ShallowNet*



Gambar 6. Ilustrasi desain arsitektur *LeNet*



Gambar 7. Ilustrasi desain arsitektur *AlexNet*

2) *LeNet*: Gambar 6 adalah ilustrasi desain arsitektur *LeNet* secara umum sedangkan pengaturan yang dilakukan pada tiap *layer*-nya dapat dilihat pada Tabel II.

TABEL II
DETAIL PENGATURAN TIAP LAYER PADA ARSITEKTUR *LENET*

Hidden Layer Ke-	Pengaturan
1	<i>Convolution Layer</i> dengan 6 <i>filter</i> berukuran 3x3 dilanjutkan dengan <i>max pooling</i> dengan matrik berukuran 2x2
2	<i>Convolution Layer</i> dengan 16 <i>filter</i> berukuran 5x5 dilanjutkan dengan <i>max pooling</i> dengan matrik berukuran 2x2
3	<i>Fully-connected Layer</i> dengan 120 <i>node</i>
4	<i>Fully-connected Layer</i> dengan 84 <i>node</i>

3) *AlexNet*: Gambar 7 adalah ilustrasi desain arsitektur *AlexNet* secara umum sedangkan pengaturan yang dilakukan pada tiap *layer*-nya dapat dilihat pada Tabel III.

TABEL III
DETAIL PENGATURAN TIAP LAYER PADA ARSITEKTUR *ALEXNET*

Hidden Layer Ke-	Pengaturan
1	<i>Convolution Layer</i> dengan 96 <i>filter</i> berukuran 11x11 dilanjutkan dengan <i>max pooling</i> dengan matrik berukuran 3x3
2	<i>Convolution Layer</i> dengan 256 <i>filter</i> berukuran 5x5 dilanjutkan dengan <i>max pooling</i> dengan matrik berukuran 3x3
3	<i>Convolution Layer</i> dengan 384 <i>filter</i> berukuran 3x3 tanpa <i>pooling</i>
4	<i>Convolution Layer</i> dengan 384 <i>filter</i> berukuran 3x3 tanpa <i>pooling</i>
5	<i>Convolution Layer</i> dengan 256 <i>filter</i> berukuran 3x3 dilanjutkan dengan <i>max pooling</i> dengan matrik berukuran 3x3
6	<i>Fully-connected Layer</i> dengan 4096 <i>node</i>
7	<i>Fully-connected Layer</i> dengan 4096 <i>node</i>
8	<i>Fully-connected Layer</i> dengan 1000 <i>node</i>

F. Implementasi Program

Bahasa pemrograman yang digunakan dalam penelitian ini adalah *Python 3*. Pengaturan arsitektur CNN beserta pelatihan dan juga pengujiannya menggunakan *library Tensorflow* dengan *Keras API*. Program dijalankan pada komputer dengan prosesor 2,3 Ghz Intel Core i5, memori dengan RAM 8GB DDR3L, dan sistem operasi *Windows 10*. *Keras API* merangkum dan menyederhanakan fungsi-fungsi yang ada dalam *library Tensorflow*. Pembuatan arsitektur CNN dalam *Keras API* dilakukan dengan membuat *Sequential* (fungsi untuk membuat model). Untuk menambah proses seperti *convolution*, *pooling*, dan *fully connected* digunakan perintah `.add` (proses yang ditambahkan pada *Sequential*). Proses konvolusi dalam *Keras* menggunakan fungsi `Conv2D()`, proses *pooling* menggunakan fungsi `MaxPooling2D()`, proses normalisasi menggunakan fungsi `BatchNormalization()`, dan proses *fully connected* menggunakan fungsi `Dense()`. Gambar 8 merupakan contoh kode pembuatan model CNN menggunakan *Keras API*.

```

1. model = Sequential()
2. model.add(Conv2D(16, kernel_size=(3, 3),
    strides=(1, 1),
    activation='relu',
    input_shape=input_shape))
3. model.add(MaxPooling2D(pool_size=(2, 2), strides=(2,
    2)))
4. model.add(Flatten())
5. model.add(Dense(4096, activation='relu'))
6. model.add(Dense(num_classes, activation='softmax'))
    
```

Gambar 8. Kode program pengaturan model CNN pada *Keras API*

G. Indikator Kinerja Arsitektur CNN

Model CNN pada penelitian ini ditujukan untuk menghasilkan prediksi nilai kandungan pigmen, oleh karena itu tidak seperti kebanyakan model CNN yang menggunakan indikator klasifikasi (seperti akurasi, spesifitas, dan sensitifitas), indikator yang digunakan adalah *Mean Square Error (MSE)* sebagaimana Persamaan 8. Nilai y_i adalah nilai sesungguhnya, nilai \hat{y}_i adalah nilai yang dihasilkan oleh CNN, dan n adalah banyaknya data. Semakin kecil nilai MSE semakin baik prediksi sebuah arsitektur CNN terhadap kandungan pigmen yang sesungguhnya.

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (8)$$

IV. HASIL DAN PEMBAHASAN

A. Hasil Pengumpulan Data

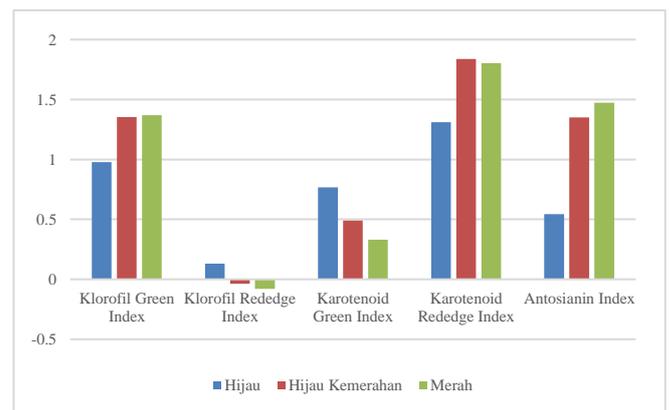
Tabel IV menunjukkan distribusi jumlah data kelompok warna visual dari keseluruhan sampel. Gambar 9 menunjukkan distribusi dari kandungan pigmen setiap kelompok warna visual pada Tabel IV untuk tanaman Pucuk Merah dan Gambar 10 menunjukkan distribusi dari

kandungan pigmen setiap kelompok warna visual pada Tabel IV untuk tanaman Sirih. Dari Gambar 9 dapat diketahui bahwa kandungan pigmen pada sampel Pucuk Merah memiliki perilaku yang sesuai dengan teori yaitu bahwa warna visual pada daun dapat menunjukkan tinggi rendahnya kandungan pigmen tertentu didalamnya. Sebagai contoh pada kelompok daun warna merah, kandungan Antosianin paling tinggi dibandingkan hijau dan hijau kemerahan.

TABEL IV
DISTRIBUSI JUMLAH DATA SAMPEL

Kelompok Warna Visual	Jumlah Data
Hijau	97
Merah	90
Kuning	50
Hijau kemerahan	147
Hijau kekuningan	7

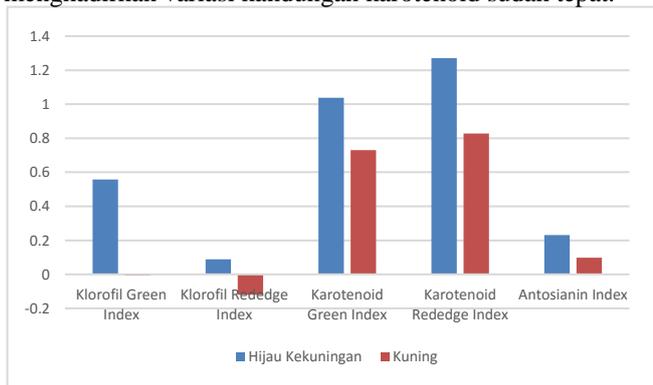
Untuk Klorofil, kandungannya pada kelompok daun warna hijau adalah yang paling tinggi. Karotenoid juga menyumbangkan warna kemerahan pada tanaman. Oleh karena itu kandungan Karotenoid pada kelompok daun warna hijau kemerahan dan merah tampak lebih tinggi jika dibandingkan dengan kelompok daun warna hijau.



Gambar 9. Distribusi kandungan pigmen tanaman Pucuk Merah

Dari Gambar 10 diketahui bahwa data sampel daun Sirih memiliki perilaku yang sesuai dengan teori sebagaimana kelompok data sebelumnya (tanaman Pucuk merah). Pada kelompok daun warna hijau kekuningan nilai klorofilnya lebih tinggi jika dibandingkan dengan kelompok daun warna kuning. Namun, untuk kandungan Karotenoid, meskipun daun terlihat cenderung hijau, ternyata kandungan Karotenoidnya lebih banyak jika dibandingkan dengan daun yang terlihat lebih kuning. Hal ini sebenarnya juga sesuai dengan teori yang menjelaskan bahwa karena Klorofil letaknya secara struktur lebih dekat dengan permukaan daun, maka Klorofil lebih mudah terlihat jika dibandingkan dengan Karotenoid. Secara umum, pada kelompok sampel daun tanaman Sirih, kandungan Karotenoid jauh lebih

banyak dibandingkan dengan kandungan Klorofilnya. Hal ini mengkonfirmasi bahwa pemilihan tanaman sirih untuk menghadirkan variasi kandungan karotenoid sudah tepat.



Gambar 10. Distribusi kandungan pigmen tanaman Sirih

Tabel V menginformasikan nilai minimum dan maksimum dari kandungan pigmen pada daun Pucuk Merah. Tabel VI menginformasikan nilai minimum dan maksimum dari kandungan pigmen pada daun Sirih. Informasi ini digunakan sebagai pertimbangan desain arsitektur CNN. Dari kedua tabel tersebut dapat dikonfirmasi bahwa daun Pucuk Merah lebih banyak kandungan antosianin-nya jika dibandingkan dengan daun Sirih sedangkan daun Sirih lebih banyak kandungan karotenoidnya jika dibandingkan dengan daun Pucuk Merah. Hal ini telah sesuai dengan tujuan pemilihan kedua tanaman tersebut, yaitu Pucuk Merah untuk menyediakan variasi data kandungan Antosianin dan Sirih untuk menyediakan variasi data kandungan Karotenoid. Kedua-duanya juga menyediakan variasi data untuk kandungan Klorofil.

TABEL V
NILAI MINIMUM DAN MAKSIMUM KANDUNGAN PIGMEN DAUN PUCUK MERAH

Indeks Kandungan Pigmen	Minimum	Maksimum
$(Chl)RI_{green}$	0.407663	1.857813
$(Chl)RI_{red}$	-0.107580	0.323876
CRI_{green}	0.157148	0.981661
CRI_{red}	0.814667	2.226155
ARI	0.283901	1.967888

TABEL VI
NILAI MINIMUM DAN MAKSIMUM KANDUNGAN PIGMEN DAUN SIRIH

Indeks Kandungan Pigmen	Minimum	Maksimum
$(Chl)RI_{green}$	-0.116920	0.793131
$(Chl)RI_{red}$	-0.161180	0.161759
CRI_{green}	0.442079	1.137226
CRI_{red}	0.511603	1.398290
ARI	0.065826	0.302499

B. Pemilihan Arsitektur CNN Terbaik

Eksperimen untuk mendapatkan arsitektur CNN terbaik dilakukan dengan cara membuat beberapa perubahan pada parameter CNN yaitu *optimizer*, *batch size*, dan *epoch*. Adam dan RMSProp dipilih sebagai variasi *optimizer*. *Batch size* dicoba dengan ukuran 30, 60, dan 120 sedangkan *epoch* dicoba sejumlah 15, 30, dan 45.

1) *ShallowNet*: Tabel VII dan VIII merangkum kinerja dari arsitektur *ShallowNet* dengan indikator MSE. Tampak bahwa arsitektur terbaik (baik saat pelatihan maupun pengujian) diperoleh dengan menggunakan *optimizer* Adam dengan *batch size* 30 (perhatikan kolom yang diberi warna abu-abu). Jumlah *epoch* yang ideal adalah 15. Meskipun saat pengujian MSE arsitektur dengan menggunakan jumlah *epoch* 30 sedikit lebih baik daripada arsitektur dengan jumlah *epoch* 15, namun dengan memilih jumlah *epoch* yang lebih sedikit sistem dapat bekerja lebih cepat.

TABEL VII
NILAI MSE ARSITEKTUR *SHALLOWNET* UNTUK DATA PELATIHAN

Epoch	Batch Size					
	30		60		120	
	Adam	RMSprop	Adam	RMSprop	Adam	RMSprop
15	0.0055	0.0488	0.0333	0.0550	0.0614	0.1204
30	0.0066	0.0224	0.0319	0.0373	0.0575	0.3457
45	0.0096	0.0172	0.0276	0.0257	0.6186	0.0606

TABEL VIII
NILAI MSE ARSITEKTUR *SHALLOWNET* UNTUK DATA UJI

Epoch	Batch Size					
	30		60		120	
	Adam	RMSprop	Adam	RMSprop	Adam	RMSprop
15	0.0290	0.0760	0.0384	0.0666	0.0481	0.1030
30	0.0234	0.0519	0.0347	0.0635	0.0587	0.3253
45	0.0378	0.0559	0.0360	0.0356	0.6257	0.0933

2) *LeNet*: Tabel IX dan X merangkum kinerja dari arsitektur *LeNet* dengan indikator MSE. Sebagaimana yang terjadi pada arsitektur *ShallowNet*, tampak bahwa arsitektur terbaik (baik saat pelatihan maupun pengujian) diperoleh dengan menggunakan *optimizer* Adam dengan *batch size* 30 (perhatikan kolom yang diberi warna abu-abu). Jumlah *epoch* yang ideal adalah 30. Meskipun saat pengujian MSE arsitektur dengan menggunakan jumlah *epoch* 45 sedikit lebih baik daripada arsitektur dengan jumlah *epoch* 15, namun dengan memilih jumlah *epoch* yang lebih sedikit sistem dapat bekerja lebih cepat.

TABEL IX
NILAI MSE ARSITEKTUR *LENET* UNTUK DATA PELATIHAN

Epoch	Batch Size					
	30		60		120	
	Adam	RMSprop	Adam	RMSprop	Adam	RMSprop
15	0.0171	0.0452	0.0239	0.0399	0.0265	0.0503
30	0.0131	0.0362	0.0195	0.0433	0.0207	0.0550
45	0.0144	0.0139	0.0181	0.0279	0.1493	0.0263

TABEL X
NILAI MSE ARSITEKTUR *LENET* UNTUK DATA UJI

Epoch	Batch Size					
	30		60		120	
	Adam	RMSprop	Adam	RMSprop	Adam	RMSprop
15	0.0234	0.0409	0.0297	0.0371	0.0238	0.0449
30	0.0218	0.0377	0.0279	0.0399	0.0288	0.0636
45	0.0214	0.0279	0.0265	0.0352	0.1482	0.0361

3) *AlexNet*: Tabel XI dan XII merangkum kinerja dari arsitektur *AlexNet* dengan indikator MSE. Berbeda dari dua arsitektur sebelumnya, tampak bahwa arsitektur terbaik (baik saat pelatihan maupun pengujian) diperoleh dengan menggunakan *optimizer* RMSProp dengan *batch size* 30 (perhatikan kolom yang diberi warna abu-abu). Jumlah *epoch* yang ideal adalah 30.

TABEL XI
NILAI MSE ARSITEKTUR *ALEXNET* UNTUK DATA PELATIHAN

Epoch	Batch Size					
	30		60		120	
	Adam	RMSprop	Adam	RMSprop	Adam	RMSprop
15	0.1495	0.1125	0.0296	0.1167	0.1487	0.0888
30	0.1528	0.0253	0.1510	0.2401	0.1177	0.0963
45	0.1523	0.0324	0.1492	0.0486	0.0686	0.1208

TABEL XII
NILAI MSE ARSITEKTUR *ALEXNET* UNTUK DATA UJI

Epoch	Batch Size					
	30		60		120	
	Adam	RMSprop	Adam	RMSprop	Adam	RMSprop
15	0.1483	0.1005	0.0324	0.1093	0.1466	0.0775
30	0.1507	0.0236	0.1500	0.2318	0.1014	0.0891
45	0.1512	0.0343	0.1478	0.0427	0.0581	0.1210

Dari keseluruhan eksperimen dengan 3 jenis arsitektur yang berbeda, maka ditetapkan bahwa untuk kasus prediksi kandungan pigmen fotosintesis ini arsitektur terbaik adalah *ShallowNet* dengan *optimizer* Adam, *batch size* 30 dan jumlah *epoch* 15. Arsitektur tersebut memberikan MSE yang paling kecil diantara arsitektur lainnya yaitu sebesar 0.0055. Semakin kompleks arsitektur CNN yang digunakan, kinerja prediksi kandungan pigmen semakin kurang baik. Susunan *ShallowNet* yang lebih sederhana dibandingkan dengan *LeNet* dan *AlexNet* ternyata mampu memodelkan hubungan antara warna visual yang terekam pada citra digital RGB dengan kandungan pigmen fotosintesis lebih baik dibandingkan dengan arsitektur yang susunannya lebih rumit.

Gambar 11 menyajikan perbandingan kinerja arsitektur CNN terbaik untuk setiap jenis pigmen dengan menggunakan data pengujian. Sumbu vertikal adalah nilai MSE. Tampak bahwa CNN mempunyai kinerja prediksi yang berbeda-beda terhadap indeks reflektansi yang digunakan untuk mengukur kandungan pigmen pada setiap kelompok warna visual daun. Untuk nilai Klorofil Green ($(Chl)RI_{green}$), CNN paling bagus dalam memprediksinya pada kelompok daun dengan warna visual hijau kemerahan namun kurang bagus pada kelompok daun hijau kekuningan. Untuk nilai Klorofil Rededge ($(Chl)RI_{rededge}$), CNN paling bagus dalam memprediksi kandungan klorofil pada kelompok daun merah.

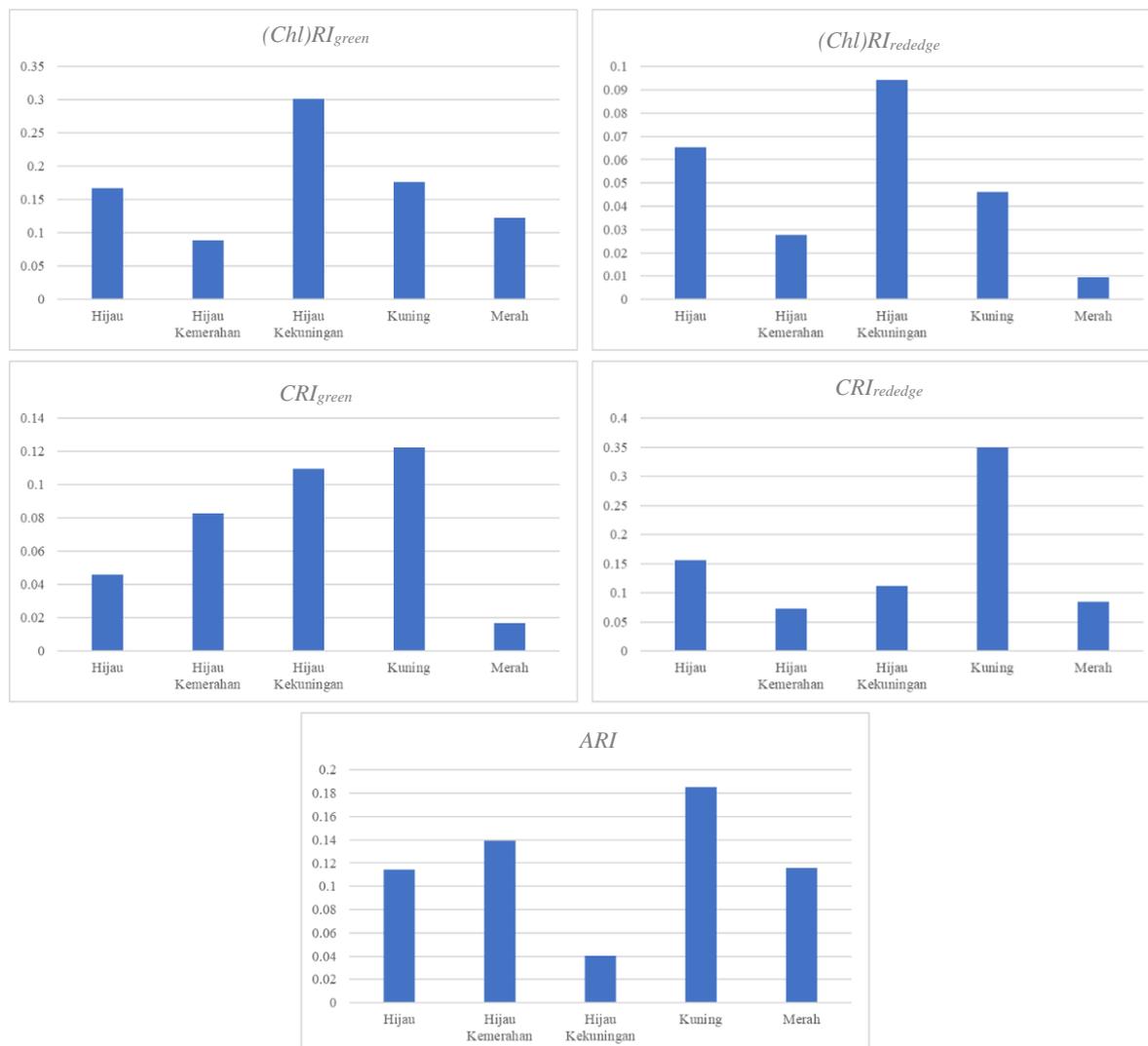
Namun, sebagaimana perilakunya untuk nilai Klorofil Green, nilai Klorofil Rededge juga kurang bagus diprediksi pada kelompok daun hijau kekuningan. Untuk nilai Karotenoid Green (CRI_{green}) dan nilai Karotenoid Rededge ($CRI_{rededge}$), CNN bagus dalam memprediksinya pada kelompok daun merah namun kurang baik pada kelompok daun kuning. Sedangkan untuk nilai Antosianin (*ARI*), CNN bagus dalam memprediksinya pada kelompok daun hijau kekuningan namun kurang baik pada kelompok daun kuning. Untuk menjelaskan fenomena tersebut perlu dilakukan penelitian lanjutan dengan penambahan keberagaman gradasi warna daun. Kondisi tersebut dapat dicapai salah satunya adalah dengan menambah jenis tanaman yang dijadikan sampel. Selain itu juga perlu dipelajari distribusi data dari tiap jenis indeks reflektansi yang kemungkinan dapat memberikan pertimbangan untuk desain fungsi aktivasi yang digunakan pada *fully connected layer*. Pada penelitian ini fungsi aktivasi yang digunakan adalah *LeakyRelu*.

C. Limit Detection

Setelah ditemukan arsitektur terbaiknya, maka langkah terakhir pada penelitian ini adalah melakukan uji coba *limit detection*. Uji coba tersebut dimaksudkan untuk mengetahui sejauh mana arsitektur CNN dapat dilatih hingga MSE-nya lebih kecil dari yang telah ditemukan sebelumnya. Untuk tujuan tersebut sistem dilatih kembali tanpa pembatasan jumlah *epoch* namun dibatasi pencapaian MSE-nya. Kemudian dipantau pergerakan nilai MSE hingga polanya

tidak lagi turun namun berubah menjadi naik. Hasilnya dapat dilihat pada Tabel XIII.

MSE cenderung semakin besar. Tampak bahwa jika sistem dipaksa untuk mencapai nilai MSE kurang dari 10^{-5} sistem tidak lagi mampu mengenali data uji dengan baik meskipun sangat baik dalam mengenali data pelatihan (*overfitting*).



Gambar 11. Grafik perbandingan kinerja prediksi arsitektur CNN terbaik untuk setiap jenis pigmen dengan menggunakan MSE

TABEL XIII
RANGKUMAN UJI COBA LIMIT DETECTION

Batas MSE	Epoch	Nilai MSE yang dicapai		Waktu Pelatihan (detik)
		Pelatihan	Pengujian	
$< 10^{-3}$	50	0.00096	0.0066	50
$< 10^{-4}$	112	0.000084	0.0069	112
$< 10^{-5}$	200	0.0000043	1.3300	200

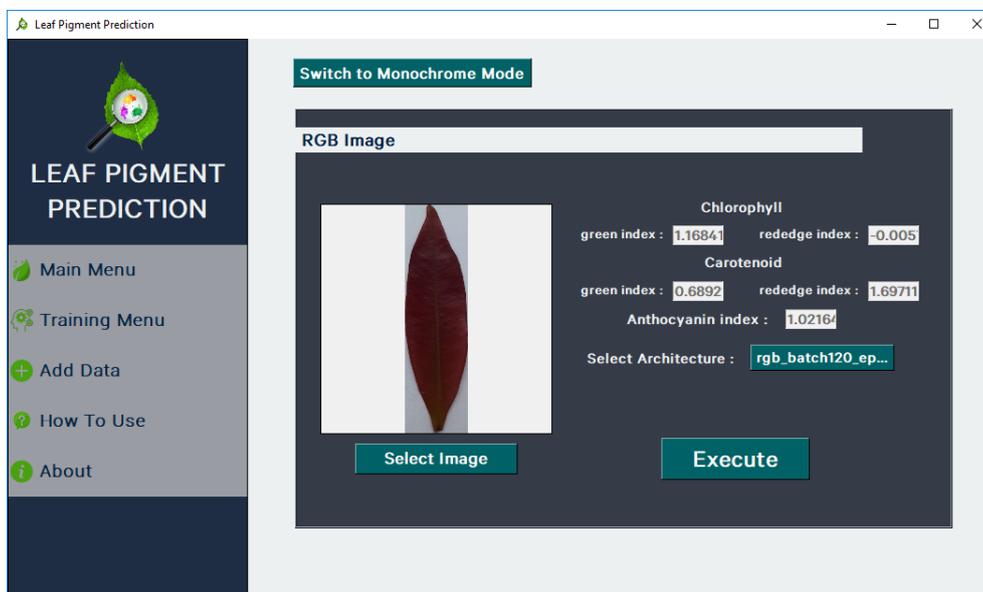
Tabel tersebut menunjukkan bahwa semakin banyak jumlah *epoch* maka MSE juga akan semakin kecil, namun pada *epoch* tertentu (kurang lebih diatas 150) nilai MSE akan menunjukkan angka yang tetap yang apabila diteruskan nilai

Fakta lain yang diperoleh adalah bahwa meskipun MSE pelatihan dapat turun terus, MSE pengujian tidak jauh berbeda dengan MSE arsitektur yang dilatih hanya menggunakan 15 *epoch*. Oleh karena itu, untuk tujuan prediksi kandungan pigmen, tidak disarankan untuk melatih arsitektur CNN secara berlebihan. Selain memakan waktu yang lama, arsitektur kehilangan kemampuan *generalisasi* sehingga tidak dapat lagi diandalkan untuk melakukan prediksi pada citra digital baru.

D. Antarmuka Aplikasi

Untuk mempermudah eksperimen, telah dikembangkan pula prototipe antarmuka aplikasi yang dapat diakses dengan menggunakan Python 3 dan *library* tkinter. Salah satu contoh tampilan antarmuka tampak pada Gambar 12.

4. Terdapat perbedaan kinerja CNN dalam memprediksi Klorofil, Karotenoid, dan Antosianin yang masih memerlukan eksperimen lanjutan untuk dapat dijelaskan dengan lebih detail



Gambar 12. Contoh tampilan prototipe antarmuka aplikasi

Laman tersebut adalah fasilitas bagi *user* untuk melakukan prediksi sampel daun yang baru. Tombol **Select Image** digunakan untuk memilih *file* gambar yang akan digunakan sebagai *input* CNN. Tombol **Execute** digunakan untuk memerintahkan CNN melakukan prediksi yang hasilnya kemudian ditampilkan pada *textbox* di sebelah kanan gambar. Melalui laman tersebut juga dapat diakses petunjuk penggunaan aplikasi dan fasilitas pelatihan untuk uji coba arsitektur baru.

V. KESIMPULAN

Beberapa kesimpulan yang didapatkan melalui eksperimen yang telah dilakukan adalah sebagai berikut:

1. Model CNN mampu menemukan fitur warna terbaik pada citra digital RGB daun dan digunakan untuk melakukan prediksi kandungan pigmen fotosintesis
2. Arsitektur dengan konsep *ShallowNet* diketahui sebagai arsitektur terbaik dengan menggunakan *optimizer* Adam, ukuran *batch* 15 yang telah dilatih dengan menggunakan 15 *epoch*
3. Arsitektur terbaik tersebut mampu memberikan MSE pelatihan sebesar 0.0055 dan MSE pengujian sebesar 0.029 pada data dengan rentang -0.1 s.d. 2.2 dan tidak terindikasi adanya *overfitting*

5. Kemampuan prediksi arsitektur CNN dapat ditingkatkan hingga mencapai MSE kurang dari 10^{-4} dengan penambahan jumlah *epoch*.

UCAPAN TERIMA KASIH

Penelitian ini terlaksana atas pembiayaan hibah Penelitian Dasar Unggulan Perguruan Tinggi (PDUPT) pendanaan tahun 2018. Terima kasih kami sampaikan kepada Kementerian Riset, Teknologi dan Pendidikan Tinggi atas fasilitasi yang diberikan melalui hibah tersebut.

DAFTAR PUSTAKA

- [1] A.J.S. Neto, D.C. Lopes, & J.C.F.B Junior, "Assesment of Photosynthetic Pigment and Water Contents in Intact Sunflower Plants from Spectral Indices," *Agriculture*, vol.7, no.8. 2017.
- [2] A.E. Strever, "Non-Destructive Assessment of Leaf Composition as Related to Growth of The Grapevine (*Vitis vinifera* L. cv. Shiraz)," Doctoral thesis, Stellenbosch University, 2012.
- [3] (2017) The Avantes website. [Online]. Tersedia: www.azom.com/article.aspx?ArticleID=14434.
- [4] L. Li, Q. Zhang, & D. Huang, "A Review of Imaging Techniques for Plant Phenotyping," *Sensors*, vol.14, pp.20078-20111, 2014.
- [5] M. Radovic, O. Adarkwa, & Q. Wang, "Object Recognition in Aerial Image Using Convolutional Neural Networks," *Journal of Imaging*, vol.3, no.21, pp.1-9, 2017.
- [6] B. Zhao, J. Feng, X. Wu, & S. Yang, "A Survey on Deep Learning-Based Fine-grained Object Classification and Semantic

- Segmentation,” *International Journal of Automation and Computing*, vol.14, no.2, pp.119-135, 2017.
- [7] Z. Cheng, X. Li, & C.C. Loy, “Pedestrian Color Naming via Convolutional Neural Network,” *Proceeding ACCV*, 2018, pp.35-51.
- [8] V.O. Yazici, J.V. Weijer, & A. Ramisa, “Color Naming for Multi-Color Fashion Items,” *Proceeding WorldCIST’18*, 2018, pp. 64-73.
- [9] M. Dyrmann, H. Karstoft, & H.S. Midtiby, “Plant Species Classification Using Deep Convolutional Neural Network,” *Biosystems Engineering*, *Engineering*, vol.151, pp.72-80, 2016.
- [10] M.M. Ghazi, B. Yanikoglu, & E. Aptoula, “Plant Identification Using Deep Neural Networks Via Optimization of Transfer Learning Parameters,” *Neurocomputing*, vol.235, pp.228-235, 2017.
- [11] S.P. Mohanty, D.P. Hughes, & M. Salathe, “Using Deep Learning for Image-Based Plant Disease Detection,” *Frontier in Plant Science*, vol.22, no.7, 2016.
- [12] E. Mlodzinska, “Survey of Plant Pigments: Molecular and Environmental Determinants of Plant Colors,” *Acta Biologica Cracoviensia*, vo. 51, no.10, pp.7-16, 2009.
- [13] A.S. Harrera, “The Biological Pigments in Plants Physiology,” *Agricultural Sciences*, vol.6, pp.1262-1271, 2015.
- [14] H. Croft, & J.M. Chen, *Leaf Pigment Content*. Elsevier Canada, 2017.
- [15] A.A. Gitelson, & M.N. Merzlyak, “Non-Destructive Assessment of Chlorophyll Carotenoid and Anthocyanin Content in Higher Plant Leaves: Principles and Algorithms,” *Proceeding Remote Sensing for Agriculture and the Environment Conf.*, Greece, Ella. 2004, pp.78-94.
- [16] Sukardi, Z. Arifin, & M. Risaldi, 2017. “Klasifikasi Penentuan Gambar Berbasis Tensorform dan Framework dengan Algoritma CNN,” *Prosiding SEMNASTIKOM*, 2017.
- [17] H.E. Khiyari, & H. Wechsler, “Face Recognition across Time Lapse Using Convolutional Neural Networks,” *Journal of Information Security*, vol.7, pp. 141-151, 2016.
- [18] S. Albewi, & A. Mahmood, “A Framework for Designing the Architectures of Deep Convolutional Neural Networks,” *Entropy*, vol.19, no. 6, pp. 1-20, 2017.