

**KLASIFIKASI SPESIES BUNGA EDELWEIS MENGGUNAKAN
CONVOLUTIONAL NEURAL NETWORK DAN PENGOPTIMAL
ADAPTIVE MOMENT ESTIMATION**

SKRIPSI

Diajukan untuk menempuh Ujian Sarjana
pada Fakultas Matematika dan Ilmu Pengetahuan Alam
Universitas Padjadjaran

**WIBI ANTO
NPM 140110200025**



**UNIVERSITAS PADJADJARAN
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
PROGRAM STUDI S-1 MATEMATIKA
JATINANGOR
2024**

|

LEMBAR PENGESAHAN

**JUDUL : KLASIFIKASI SPESIES BUNGA EDELWEIS
MENGUNAKAN CONVOLUTIONAL NEURAL
NETWORK DAN PENGOPTIMAL ADAPTIVE MOMENT
ESTIMATION**

**PENULIS : WIBI ANTO
NPM : 140110200025**

Jatinangor, Maret 2024

Menyetujui,
Pembimbing Utama Pembimbing Pendamping

Herlina Napitupulu M.Sc., Ph.D.
NIP 198804082019032015

Nurul Gusriani S.Si., M.Si.
NIP 197008181998032001

Mengetahui,
Ketua Program Studi S-1 Matematika
Fakultas Matematika dan Ilmu Pengetahuan Alam
Universitas Padjadjaran

Edi Kurniadi, S.Si., M.Si., Ph.D
NIP 198104182008121003

ABSTRAK

Wibi Anto
NPM 140110200025

Indonesia adalah negara tropis yang memiliki banyak kekayaan hayati. Pemerintah mengatur perlakuan tumbuhan yang dilindungi untuk menjaga kelestariannya. Salah satu karakteristik hewan dan tumbuhan dilindungi adalah endemik, yaitu suatu organisme hanya dapat ditemukan di wilayah tertentu. *Anaphalis javanica* merupakan salah satu spesies Edelweis endemik yang ada di Indonesia. Selain *Anaphalis javanica*, banyak spesies lain dari Edelweis yang memiliki kemiripan dari segi warna dan bentuk. Membedakan spesies bunga Edelweis menjadi salah satu masalah yang rumit terutama bagi kalangan awam. Permasalahan tersebut dapat diatasi dengan melakukan penelitian klasifikasi Edelweis menggunakan *Convolutional Neural Network* dengan pengoptimal Adam. Penelitian dilakukan menggunakan sebanyak 3500 data gambar *train* yang kemudian akan dibagi ke dalam data *train* dan *validation* serta 1050 data gambar test yang terbagi ke dalam tiga kelas. Kelas data dalam penelitian ini merupakan nama-nama spesies Edelweis yang ada pada dataset yaitu *Anaphalis javanica*, *Leontopodium alpinum*, dan *leucogenes grandiceps*. Penelitian ini bertujuan untuk mengetahui penggunaan algoritma Adam sebagai pengoptimal dalam permasalahan klasifikasi gambar, memperoleh hasil klasifikasi bunga Edelweis menggunakan CNN, dan mengetahui pengaruh pemilihan nilai *learning rate* terhadap hasil akurasi model.

Kata kunci: Klasifikasi Gambar; Bunga Edelweis; CNN; Pengoptimal Adam; Konvolusi.

ABSTRACT

Wibi Anto
NPM 140110200025

Indonesia is a tropical country that has a lot of biological wealth. In an effort to maintain its sustainability, the government regulates the treatment of protected plants with one of its characteristics being endemic. Anaphalis javanica is one of the endemic edelweiss species in Indonesia. Besides Anaphalis javanica, there are many other species of edelweiss that have similarities in terms of color and shape. Differentiating edelweiss flower species is one of the complicated problems, especially for the layman. To try to overcome this problem, edelweiss classification research will be conducted using Convolutional Neural Network with Adam optimizer. The research is conducted using 3500 train image data which will then be divided into train and validation data and 1050 test image data which are divided into three classes. The data classes in this study are the names of edelweiss species in the dataset, namely Anaphalis javanica, Leontopodium alpinum, and Leucogenes grandiceps. This study aims to determine the use of Adam's algorithm as an optimizer in image classification problems, obtain the results of edelweiss flower classification using CNN, and determine the effect of selecting the learning rate value on the accuracy of the model.

Keywords: *Image Classification; Edelweis flower; CNN; Adam optimizer; Convolution.*

KATA PENGANTAR

Segala puji dan syukur kepada Allah SWT, karena berkat rahmat, petunjuk, dan karunia-Nya, penulis dapat menyelesaikan skripsi yang berjudul “Klasifikasi Spesies Bunga Edelweis Menggunakan *Convolutional Neural Network* dan Pengoptimal *Adaptive Moment Estimation*” Skripsi ini disusun untuk memenuhi salah satu syarat ujian sarjana pada Program Studi S-1 Matematika di Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Padjadjaran.

Skripsi ini tidak dapat selesai tanpa adanya dukungan dan bantuan dari berbagai pihak. Oleh karena itu pada kesempatan ini, penulis mengucapkan terima kasih kepada Ibu Herlina Napitupulu M.Sc., Ph.D., selaku Dosen Pembimbing Utama dan Ibu Nurul Gusriani S.Si., M.Si., selaku Dosen Pembimbing Pendamping yang telah memberikan bimbingan, bantuan, dan dorongan yang sangat berharga kepada penulis dalam proses penyusunan skripsi ini. Selain itu, penulis juga ingin mengucapkan terima kasih kepada:

1. Kedua orang tua dan kakak-kakak penulis yang selalu memberikan dukungan penuh, motivasi serta doa yang tidak pernah terputus kepada penulis.
2. Prof. Dr. Iman Rahayu, S.Si., M.Si., selaku Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Padjadjaran.
3. Dr. Ema Carnia, M.Si., selaku Kepala Departemen Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Padjadjaran.
4. Edi Kurniadi, S.Si., M.Si., Ph.D., selaku Ketua Program Studi S-1 Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Padjadjaran.

5. Dr. Alit Kartiwa S.Si., M.Si., selaku Dosen Wali penulis.
6. Seluruh Civitas Akademika Departemen Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Padjadjaran.
7. Seluruh pihak yang tidak dapat penulis sebutkan satu persatu yang telah memberikan banyak dukungan dan doa kepada penulis.

Semoga skripsi ini dapat bermanfaat bagi diri penulis sendiri, orang-orang yang membacanya, dan bahkan masyarakat secara luas lewat segala ilmu dan gagasannya.

Jatinangor, Maret 2024

Penulis

DAFTAR ISI

ABSTRAK.....	iii
<i>ABSTRACT</i>	iv
KATA PENGANTAR.....	v
DAFTAR ISI.....	vii
DAFTAR TABEL.....	ix
DAFTAR GAMBAR.....	x
DAFTAR LAMPIRAN.....	xi
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Identifikasi Masalah.....	5
1.3 Batasan Masalah.....	5
1.4 Tujuan Penelitian.....	6
1.5 Kegunaan Penelitian.....	6
1.6 Metodologi Penelitian.....	6
1.7 Sistematika Penulisan.....	7
BAB II LANDASAN TEORI.....	8
2.1 Klasifikasi Gambar.....	8
2.2 Pra-pemrosesan gambar.....	8
2.3 <i>Convolutional Neural Network</i>	9
2.4 <i>Convolution Layer</i> dua dimensi.....	10
2.5 <i>Pooling layer</i> dua dimensi.....	12
2.6 <i>Flatten layer</i>	13
2.7 <i>Dense layer</i>	13
2.8 Pengoptimal.....	15
2.9 Metrik <i>Accuracy</i>	18

2.10	TensorFlow	19
2.11	Matplotlib.....	19
2.12	<i>Google Colaboratory</i>	19
BAB III OBJEK DAN METODE PENELITIAN		20
3.1	Objek Penelitian.....	20
3.2	Metode Penelitian	20
3.3	Diagram Alir Penelitian	24
DAFTAR PUSTAKA		29
RIWAYAT HIDUP PENULIS		31

DAFTAR TABEL

Tabel 1. Judul Tabel.....	XX
Tabel 2. Judul Tabel.....	XX
Tabel 3. Judul Tabel.....	XX
Tabel 4. Judul Tabel.....	XX

DAFTAR GAMBAR

Gambar 2.1. Arsitektur CNN secara umum	10
Gambar 2.2 Tipe arsitektur CNN dengan dua tahap ekstraksi fitur.....	10
Gambar 2.3 <i>Hyperparameter</i> dalam fungsi Conv2D.....	11
Gambar 2.4 Ilustrasi <i>max pooling</i> 2D dengan input 4×4 dan <i>pool size</i> (2,2)	13
Gambar 2.5 Ilustrasi <i>flatten layer</i> dengan dimensi input (2, 2, 1)	13
Gambar 2.6 Perbandingan Adam dengan pengoptimal lainnya.....	16
Gambar 2.7. Perbandingan pemilihan <i>learning rate</i>	18
Gambar 3.1 Arsitektur model CNN	22
Gambar 3.2 Diagram alir penelitian.....	24
Gambar 3.3 Diagram alir tahap kompresi gambar	25
Gambar 3.4 Diagram alir pada tahap <i>training</i> model	26
Gambar 3.5 Diagram alir pada tahap ekstraksi fitur	27
Gambar 3.6 Diagram alir pada tahap klasifikasi di lapisan <i>Fully-connected Layer</i>	28

DAFTAR LAMPIRAN

Lampiran 1. Nama Lampiran	XX
Lampiran 2. Nama Lampiran	XX
Lampiran 3. Nama Lampiran	XX
Lampiran 4. Nama Lampiran	XX

BAB I

PENDAHULUAN

1.1 Latar Belakang

Indonesia adalah negara tropis yang memiliki banyak kekayaan alam termasuk kekayaan hayati. Sebagai upaya pelestarian dari beragamnya kekayaan alam di Indonesia, pemerintah melalui PP No.5 Tahun 1990 membagi kawasan pelestarian menjadi taman nasional, taman hutan raya, dan taman wisata alam. Taman Nasional adalah kawasan pelestarian alam yang mempunyai ekosistem asli, dikelola dengan sistem zonasi yang dimanfaatkan untuk tujuan penelitian, ilmu pengetahuan, pendidikan, menunjang budidaya, pariwisata, dan rekreasi.

Setiap Taman Nasional memiliki karakteristik yang berbeda bergantung pada lokasi geografis. Perbedaan karakteristik tersebut diantaranya adalah ragam keanekaragaman hayati yang juga didukung dengan perbedaan topografi dari Taman Nasional. Taman Nasional dengan topografi pegunungan cenderung memiliki keragaman hayati yang mampu beradaptasi dengan ketinggian dan suhu yang rendah sedangkan Taman nasional dengan topografi pesisir cenderung memiliki karakteristik hayati yang mampu beradaptasi dan berinteraksi dengan ekosistem laut.

Perbedaan karakteristik Taman Nasional membuat adanya beberapa hayati yang hanya dapat tumbuh pada lokasi tertentu, salah satunya adalah bunga Edelweis. Menurut Direktorat Jenderal Konservasi Sumber Daya Alam dan Ekosistem, bunga Edelweis hanya dapat tumbuh di daerah pegunungan dan memerlukan sinar matahari penuh. Pemerintah melalui Permen LHK No. P.106/MENLHK/SETJEN/KUM.1/12/2018 menyatakan salah satu spesies endemik bunga Edelweis yaitu *Anaphalis javanica* sebagai jenis tumbuhan yang dilindungi. Bunga Edelweis telah memenuhi kriteria sebagai tumbuhan yang dilindungi. Berdasarkan PP No.7 Tahun 1999 Pasal 5 ayat (1), kriteria tumbuhan dan satwa yang dilindungi adalah: mempunyai populasi yang kecil, adanya penurunan yang tajam pada jumlah individu di alam, dan daerah penyebaran yang terbatas.

Bunga Edelweis memiliki jenis spesies yang sangat beragam, dengan bentuk dan warna yang serupa sehingga cukup sulit untuk membedakannya secara visual. Hal ini menghadirkan tantangan dalam identifikasi spesies bunga Edelweis secara akurat. Salah satu bidang *Machine Learning* yaitu *Computer Vision* mampu menjawab tantangan tersebut. Salah satu cabang *computer vision* yaitu *Image Classification* atau klasifikasi gambar dengan menggunakan *Convolutional Neural Network* (CNN) mampu melakukan pengenalan pola dan fitur visual yang rumit, sehingga memungkinkan untuk membedakan spesies-spesies bunga Edelweis dengan lebih tepat.

Klasifikasi gambar menggunakan CNN secara umum memiliki dua buah tahapan pelatihan model yaitu proses ekstraksi fitur dan proses klasifikasi (Géron, 2019). Proses ekstraksi fitur bertujuan untuk memperoleh fitur yang ada pada gambar dan proses tersebut dilakukan dengan menggunakan operasi konvolusi dua dimensi dan operasi *max pooling* dua dimensi. Setelah mendapatkan hasil ekstraksi, selanjutnya dilakukan proses *flatten* untuk mengubah dimensi dari hasil ekstraksi fitur menjadi satu dimensi. Kemudian dilanjutkan dengan proses klasifikasi pada lapisan *fully-connected layer* berupa *dense layer*.

Penelitian mengenai klasifikasi gambar menggunakan telah banyak dilakukan dengan berbagai objek. Alkaff & Prasetyo (2022) mengaplikasikan CNN untuk klasifikasi penyakit tanaman tomat yang menyerang daun. Penelitian tersebut menggunakan 10.000 yang kemudian dibagi dengan distribusi 80% untuk data *train* dan 20% untuk data *validation* serta 1000 data *test*. Model CNN dibuat dengan mengoptimasi *hyperparameter* menggunakan Hyperband dan mencapai tingkat akurasi sebesar 95,69% pada proses *train*, 88,5% pada proses *validation*, dan 88,6% pada proses *test*.

(Muhammad & Wibowo, 2021) menggunakan CNN dengan arsitektur ResNet50v2 untuk mengklasifikasikan empat jenis tanaman *Aglaonema* menggunakan 1960 gambar. Gambar tersebut terbagi secara merata sebanyak 440 gambar untuk data *train* dan 50 gambar untuk data *test* per jenis tanaman. Eksperimen dalam penelitian ini dilakukan dengan mencoba melakukan training dengan menghilangkan *background* gambar, mengubah *learning rate*, dan

mengubah nilai parameter *dropout* dan menghasilkan akurasi tertinggi pada proses *testing* sebesar 99% menggunakan gambar dengan *background* dan 71% menggunakan gambar tanpa *background*.

Penelitian klasifikasi bunga Edelweis juga telah dilakukan oleh Malau & Mulyana (2022) dengan mengklasifikasikan dua jenis Edelweis yaitu *Anaphalis javanica* dan *Leontopodium alpinum* menggunakan metode *Linear Discriminant Analysis* (LDA). Model yang dibuat dalam penelitian tersebut mencapai tingkat akurasi 100% dengan menggunakan 1500 data gambar *train* pada saat proses *training* dan 99,77% dengan menggunakan 450 data gambar *test* pada proses *testing*.

Pada Penelitian ini akan dilakukan klasifikasi gambar Edelweis seperti yang dilakukan Malau & Mulyana, (2022). Perbedaan penelitian ini dengan Malau & Mulyana, (2022) adalah penggunaan metode CNN dengan pengoptimal Adam. Jumlah kelas yang digunakan pada penelitian ini juga lebih banyak dibandingkan dengan Malau & Mulyana, (2022) dengan tiga buah kelas berupa jenis bunga Edelweis yaitu *Anaphalis javanica*, *Leontopodium alpinum*, dan *leucogenes grandiceps*.. Penelitian ini akan menggunakan 3500 data gambar *train* yang kemudian akan dibagi ke dalam data *train* dan *validation* serta 1050 data gambar *test*.

1.2 Identifikasi Masalah

Adapun identifikasi masalah pada penelitian ini adalah sebagai berikut:

1. Bagaimana mengimplementasikan (algoritma) CNN dengan pengoptimal *Adaptive Moment Estimation* dalam mengklasifikasikan spesies bunga Edelweis.
2. Bagaimana hasil klasifikasi spesies bunga Edelweis menggunakan model *CNN* dengan pengoptimal Adam?
3. Berapa nilai learning rate terbaik yang menghasilkan nilai akurasi tertinggi?

1.3 Batasan Masalah

Batasan permasalahan dalam penelitian ini adalah:

1. Klasifikasi spesies bunga Edelweis berdasarkan gambar bunga dilakukan menggunakan model *Machine Learning* yaitu *Convolutional Neural Network* dan pengoptimal menggunakan *Adaptive Moment Estimation* (Adam).
2. Performa model *Machine Learning* dianalisa menggunakan metrik akurasi.
3. Data yang digunakan dalam penelitian ini adalah gambar-gambar dari bunga Edelweis yang diambil dari *Kaggle* yang terdiri dari tiga spesies bunga Edelweis yang berbeda.
4. Bahasa pemrograman yang digunakan adalah bahasa pemrograman *Python* menggunakan *Google Colaboratory*.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah sebagai berikut:

1. Memperoleh/membangun algoritma CNN dengan pengoptimal *Adaptive Moment Estimation* untuk mengklasifikasikan spesies bunga Edelweis.
2. Memperoleh hasil klasifikasi dan performa akurasi model.
3. Memperoleh nilai learning rate terbaik yang menghasilkan nilai akurasi terbesar.

1.5 Kegunaan Penelitian

Kegunaan dari penelitian ini adalah sebagai berikut:

1. Model klasifikasi spesies bunga Edelweis menggunakan *Convolutional Neural Network* dan pengoptimal Adam dapat digunakan untuk mengetahui spesies bunga Edelweis hanya dengan menggunakan gambar.
2. Alat bantu untuk pembaca khususnya para pendaki maupun masyarakat umum dalam mengenali dan mengidentifikasi bunga Edelweis guna mendukung pelestarian bunga Edelweis.

1.6 Metodologi Penelitian

Metodologi Penelitian terdiri dari studi literatur dan eksperimental:

1. Studi literatur dilakukan dengan mempelajari teori terkait *Computer Vision*, klasifikasi gambar, algoritma Adam, dan *Convolutional Neural Network* dari berbagai sumber berupa buku, jurnal, dan artikel yang tersedia secara daring.

2. Studi Eksperimental dilakukan dengan membuat arsitektur model CNN dengan pengoptimal Adam serta mencari nilai parameter learning rate terbaik untuk klasifikasi jenis bunga Edelweis menggunakan bahasa pemrograman *Python* pada *Google Colaboratory*.

1.7 Sistematika Penulisan

Sistematika penulisan pada skripsi ini adalah sebagai berikut:

BAB I PENDAHULUAN, pada bab ini dijelaskan mengenai latar belakang, identifikasi masalah, batasan masalah, tujuan penelitian, kegunaan penelitian, metodologi penelitian, dan sistematika penulisan.

BAB II LANDASAN TEORI, pada bab ini dijelaskan mengenai teori-teori yang menjadi acuan dasar dalam penelitian yaitu TensorFlow, Matplotlib, *Google Colaboratory*, pra-pemrosesan gambar, *Convolution Neural Network*, *convolution layer* dua dimensi, *pooling layer* dua dimensi, *flatten layer*, *dense layer*, pengoptimal, dan metrik *accuracy*.

BAB III OBJEK DAN METODE PENELITIAN, pada bab ini berisi tentang objek penelitian, metode penelitian, dan alur penelitian.

BAB IV HASIL DAN PEMBAHASAN, pada bab ini berisi pengolahan data gambar dan hasil penelitian yang dilakukan.

BAB V SIMPULAN DAN SARAN, pada bab ini berisi simpulan dan saran dari pembahasan penelitian yang telah dilakukan untuk peneliti selanjutnya.

BAB II

LANDASAN TEORI

2.1 Klasifikasi Gambar

Klasifikasi gambar atau *Image Classification* adalah proses kategorisasi dan pemberian label dari sekumpulan piksel atau vektor dari suatu gambar berdasarkan aturan tertentu (Shinozuka & Mansouri, 2009). Terdapat dua jenis metode klasifikasi yaitu *supervised* dan *unsupervised*. Metode *unsupervised* adalah proses klasifikasi yang sepenuhnya otomatis tanpa menggunakan *training data* sedangkan metode *supervised* adalah proses klasifikasi menggunakan *training data* dan mengelompokkan ke dalam kategori yang telah dipilih sebelumnya (Shinozuka & Mansouri, 2009).

2.2 Pra-pemrosesan gambar

Pra-pemrosesan gambar atau *Image Preprocessing* adalah suatu metode untuk mengubah data gambar mentah yang mengandung *noise* dan nilai yang tidak sesuai menjadi data gambar yang bersih dan siap digunakan (Chaki & Dey, 2018). Chaki & Dey (2018) menyatakan bahwa *noise* ataupun nilai yang tidak sesuai dapat diatasi dengan beberapa cara diantaranya: *image correction*, *image enhancement*, *image restoration*, dan *image compression*. *Image compression* atau kompresi gambar digunakan untuk mengurangi ukuran gambar baik dengan mengurangi dimensi maupun kualitas gambar. Salah satu cara untuk menurunkan kualitas gambar dan menyeragamkan ukuran piksel adalah dengan menggunakan *library*

Pillow. Algoritma untuk melakukan kompresi gambar dengan menggunakan *library Pillow* dinyatakan sebagai berikut.

1. Inisiasi lebar gambar baru dalam satuan piksel = 512 px
2. Ambil ukuran lebar dan tinggi gambar asli
3. Hitung rasio gambar asli
4. Cari tinggi gambar baru dengan membagi lebar gambar baru dengan rasio gambar asli
5. Simpan dengan kualitas gambar sebesar 85%

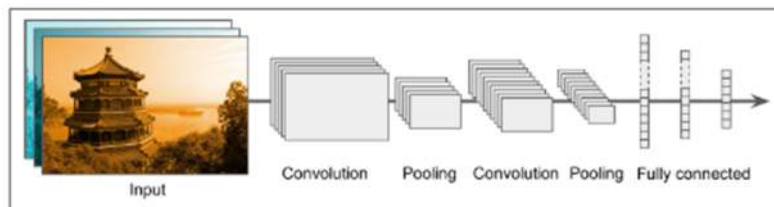
Selain menggunakan *library Pillow*, Pra-pemrosesan gambar juga dapat dilakukan menggunakan TensorFlow berupa penyeragaman dimensi, normalisasi nilai piksel, pembagian data *train* dan *validation*, pembagian *batch size*, dll. Seluruh proses ini dapat dilakukan menggunakan fungsi *ImageDataGenerator* dan beberapa metode dalam fungsi tersebut sesuai dengan kebutuhan seperti *flow_from_directory* (TensorFlow Developer, 2024b).

2.3 Convolutional Neural Network

Komputer mengenal sebuah gambar sebagai suatu matriks dengan entri berupa angka yang merepresentasikan warna dari setiap pixel-nya. Pada proses melakukan klasifikasi gambar, angka-angka pada matriks tersebut dipelajari oleh mesin untuk mencari pola tertentu. Salah satu cara untuk mempelajari pola tersebut adalah menggunakan Convolutional Neural Network.

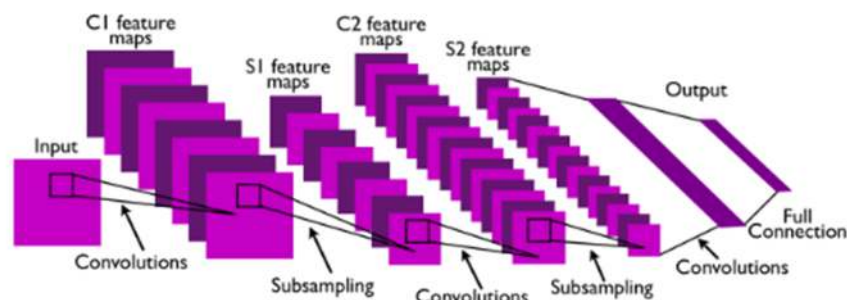
Convolutional Neural Network adalah suatu jaringan syaraf tiruan dengan proses konvolusi yang muncul dari studi tentang visual korteks otak dan telah digunakan mulai dari tahun 1980an (Géron, 2019). Menurut Ketkar (2017), Proses

konvolusi merupakan suatu perhitungan matematika yang digunakan untuk melakukan ekstraksi fitur pada gambar. Secara umum, arsitektur CNN dapat dilihat pada Gambar 2.1.



Gambar 2.1 Arsitektur CNN secara umum (Géron, 2019)

Convolutional Neural Network pada umumnya terdiri dari tiga jenis layer yaitu *Convolution Layer*, *Pooling Layer*, dan *Fully-connected Layer* (Géron, 2019). Menurut LeCun et al., (2010) *Convolution Network* merupakan arsitektur multistage yang dapat dilatih dengan input dan output dari setiap tahap berupa sekumpulan *array* yang dinamakan *feature map* dan diikuti oleh modul klasifikasi.



Gambar 2.2 Tipe arsitektur CNN dengan dua tahap ekstraksi fitur (LeCun et al., 2010).

2.4 *Convolution Layer* dua dimensi

Convolution Layer dua dimensi merupakan lapisan neuron dengan operasi konvolusi. Operasi konvolusi dapat dilakukan menggunakan beberapa alat salah satunya adalah menggunakan *python*. Pada *python*, membuat *convolution layer*

dapat menggunakan *library* TensorFlow yang didalamnya telah menyediakan fungsi Conv2D. Pada fungsi Conv2D terdapat beberapa parameter yang dapat diubah diantaranya adalah *filters*, *kernel_size*, dan *activation*. *Filters* merupakan argumen yang menyatakan banyak matriks yang digunakan dalam operasi konvolusi untuk mengekstraksi fitur. Dimensi dari matriks tersebut dinamakan *kernel size*, beberapa *kernel size* yang sering digunakan adalah (3,3), (5,5), dan (7,7).

```
tf.keras.layers.Conv2D(
    filters,
    kernel_size,
    strides=(1, 1),
    padding='valid',
    data_format=None,
    dilation_rate=(1, 1),
    groups=1,
    activation=None,
    use_bias=True,
    kernel_initializer='glorot_uniform',
    bias_initializer='zeros',
    kernel_regularizer=None,
    bias_regularizer=None,
    activity_regularizer=None,
    kernel_constraint=None,
    bias_constraint=None,
    **kwargs
)
```

Gambar 2.3 *Hyperparameter* dalam fungsi Conv2D

(TensorFlow Developer, 2024a)

Layer pertama dari *Convolution Layer* memiliki tugas tambahan yaitu memproses masukan berupa *Input Layer*. Kemudian dilakukan Operasi konvolusi dan menghasilkan *feature map* hasil konvolusi yang kemudian akan diproses dengan fungsi aktivasi pada parameter *activation*. Secara matematis, operasi konvolusi dua dimensi dengan satu filter dan satu saluran warna dinyatakan sebagai berikut.

$$y_i = \sum_a \sum_b I(a, b) * K(m - a, n - b) \quad (2.1)$$

dimana y_i merupakan elemen *feature map* ke- i , I merupakan input dengan dimensi (m, n) dan K merupakan *filter* dengan dimensi (a, b) . *Rectified Linear Unit* (ReLU) merupakan fungsi aktivasi yang sering digunakan dalam *Convolution Layer* (Ketkar, 2017). ReLU memproses nilai y_i dengan mengubah setiap nilai yang bernilai negatif menjadi nol.

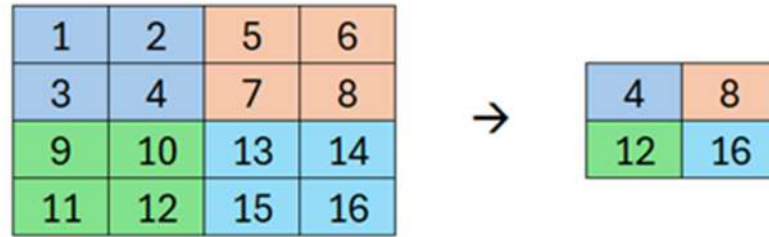
Permasalahan klasifikasi gambar dengan warna (RGB) akan mengakibatkan dimensi matriks gambar input bertambah dengan adanya elemen saluran warna. Secara matematis, proses konvolusi dua dimensi dengan satu filter dan c saluran warna dinyatakan sebagai berikut.

$$y_i = \sum_a \sum_b \sum_c I(a, b, c) * K(m - a, n - b, p - c) \quad (2.2)$$

dimana y_i merupakan elemen *feature map* ke- i , I merupakan input dengan dimensi (m, n, p) dan K merupakan *filter* dengan dimensi (a, b, c) .

2.5 *Pooling layer* dua dimensi

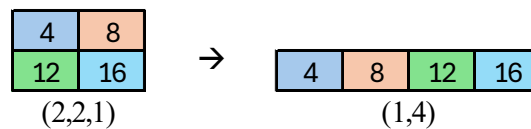
Pooling Layer adalah suatu lapisan neuron dengan operasi penggabungan atau *pooling*. Operasi *pooling* dilakukan untuk memperkecil dimensi dengan menggabungkan nilai pada area tertentu sesuai dengan ukuran filter. Terdapat dua buah jenis operasi penggabungan yaitu *max pooling* dan *average pooling*. Operasi *pooling* dapat dilakukan dengan bantuan *python* pada *library* TensorFlow yang telah menyediakan fungsi *MaxPooling2D* dengan parameter masukan *pool size* berupa *tuple* yang menyatakan dimensi *filter*.



Gambar 2.4 Ilustrasi *max pooling* 2D dengan input (4, 4) dan *pool size* (2, 2)

2.6 *Flatten layer*

Flatten layer adalah adalah suatu lapisan neuron yang memiliki operasi untuk membuat datar atau *flatten* dimensi dari tensor input yang masuk ke dalam *layer* tersebut. Misalkan terdapat I sebuah tensor dengan dimensi (W, H, C) , maka output dari dari *layer flatten* adalah $X = [I[1,1,1], I[1,2,1], \dots, I[W, H, C]]$ dengan output dimensi $(1, W \times H \times C)$.



Gambar 2.5 Ilustrasi *flatten layer* dengan dimensi input (2, 2, 1)

2.7 *Dense layer*

Dense layer atau sering juga disebut sebagai *fully-connected layer* dalam arsitektur CNN berfungsi sebagai modul pengklasifikasi hasil konvolusi (LeCun et al., 2010). TensorFlow telah menyediakan fungsi *Dense* untuk membuat *fully-connected layer*. Terdapat dua parameter yang sering diubah dalam pembuatan model CNN yaitu *units* dan *activation*. *Units* merujuk pada banyaknya sel neuron yang terdapat pada satu layer tersebut dan *activation* merujuk pada fungsi aktivasi

yang akan digunakan. Secara matematis, *output* dari *fully-connected layer* dinyatakan sebagai berikut.

$$z_i^{(L)} = \sum_j w_{ij}^{(L)} \times x_j^{(L-1)} + b_i^{(L)} \quad (2.3)$$

dimana $z_i^{(L)}$ adalah nilai *neuron* ke- i pada *layer* L , w adalah bobot penghubung antara *layer* L dan *layer* $(L - 1)$, $x_j^{(L-1)}$ adalah nilai *neuron* ke- j dari *output layer* sebelumnya dan $b_i^{(L)}$ adalah bias pada *neuron* ke- i pada *layer* L .

1. Fungsi Aktivasi *Rectified Linear Unit*

Fungsi aktivasi dalam CNN adalah fungsi matematika yang diterapkan pada keluaran setiap *neuron* dalam jaringan. Fungsi ini menentukan apakah *neuron* harus diaktifkan atau tidak berdasarkan masukan yang diterimanya. Tujuan dari fungsi aktivasi adalah untuk memperkenalkan non-linieritas ke dalam keluaran *neuron*, sehingga memungkinkan jaringan untuk belajar dan memodelkan pola kompleks dalam data. Non-linieritas ini sangat penting agar jaringan dapat belajar dan membuat prediksi yang akurat. Fungsi aktivasi yang umum digunakan dalam CNN meliputi *Rectified Linear Unit* (ReLU) atau unit linear tereduksi, yang banyak digunakan karena kesederhanaan komputasinya dan efektivitas dalam pelatihan, dan fungsi aktivasi linear. Notasi matematis dari *layer dense* dengan fungsi aktivasi *Rectified Linear Unit* (ReLU) dinyatakan sebagai berikut.

$$\alpha(z_i) = ReLU(z_i) = \max(0, z_i) \quad (2.4)$$

2. Fungsi Aktivasi *Softmax*

Fungsi aktivasi softmax adalah suatu fungsi aktivasi layer pada *neural network* dengan menggunakan regresi *softmax*. Regresi *softmax* adalah perluasan dari regresi logistik dengan kemungkinan nilai output sebanyak n nilai. Pada kasus *klasifikasi gambar*, regresi *softmax* digunakan untuk melakukan klasifikasi dengan jumlah kelas lebih dari dua (n kelas). Persamaan *Regresi softmax* dengan output sebanyak n kemungkinan adalah sebagai berikut.

$$z_i = w_i x + b_i, i = 1..n \quad (2.5)$$

maka peluang bahwa suatu nilai termasuk ke dalam kelas ke- i adalah

$$a_i = P(y = i|x) = \frac{e^{z_i}}{\sum_k^n e^{z_k}}, k = 1..n \quad (2.6)$$

dengan loss sebesar

$$loss(a_1, a_2, \dots, a_n, y) = \begin{cases} -\log a_1, y = 1 \\ -\log a_2, y = 2 \\ \vdots \\ -\log a_n, y = n \end{cases} \quad (2.7)$$

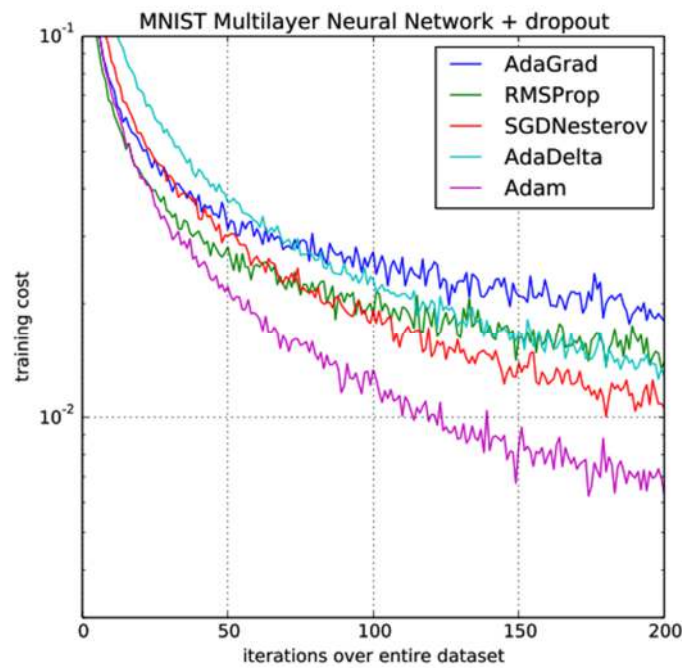
Notasi matematis dari *layer dense* dengan aktivasi *softmax* dinyatakan sebagai berikut.

$$\alpha(z_i) = Softmax(z_i) = \frac{1}{1+e^{-z_i}} \quad (2.8)$$

2.8 Pengoptimal

Pengoptimal merupakan sebuah fungsi dengan tujuan meminimumkan *loss function* dengan mengatur parameter bias dan bobot. Terdapat beberapa pengoptimal yang sering digunakan dalam *neural network* diantaranya adalah *Stochastic Gradient Descend (SGD)*, *Root Mean Square Propagation (RMSprop)*,

Adaptive Gradient (AdaGrad), *Adaptive Moment Estimation (Adam)*, dll (Chauchan, 2020). Kingma & Ba (2017) menyatakan bahwa *Adam* kuat dan cocok untuk berbagai macam masalah optimasi non-konveks di bidang *machine learning* dengan keandalannya yang mampu meminimumkan nilai dari fungsi biaya lebih cepat dibandingkan pengoptimal lainnya (Kingma & Ba, 2017).



Gambar 2.6 Perbandingan Adam dengan pengoptimal lainnya (Kingma & Ba, 2017)

1. *Categorical Cross-entropy*

Categorical cross-entropy adalah fungsi *loss* yang digunakan untuk klasifikasi multi-kelas. Fungsi ini merupakan penggabungan antara fungsi aktivasi *softmax* dan fungsi *cross-entropy loss*.

$$CE = \frac{1}{n} \sum_i^n (-\log(\alpha_i)) \quad (2.9)$$

substitusi (2.3) ke (2.4), diperoleh fungsi *categorical cross-entropy*

$$CC = \frac{1}{n} \sum_i^n \left(-\log \left(\frac{e^{z_i}}{\sum_k^n e^{z_k}} \right) \right), k = 1..n \quad (2.10)$$

2. Algoritma *Adaptive Moment Estimation*

Adaptive Moment Estimation atau Adam merupakan suatu algoritma pengoptimal berbasis stokastik gradien yang efisien karena hanya membutuhkan order pertama dari gradien dengan memakan memori yang minimal (Kingma & Ba, 2017). Misal terdapat fungsi $f(\theta)$ yang merupakan fungsi yang diferensiabel terhadap komponen stokastik θ maka aturan pembaruan adam adalah sebagai berikut.

$$g_t = \nabla_{\theta} f_t(\theta_{t-1}) \quad (2.11)$$

$$m_t = \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t \quad (2.12)$$

$$v_t = \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t^2 \quad (2.13)$$

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t} \quad (2.14)$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t} \quad (2.15)$$

$$\theta_t = \theta_{t-1} - \frac{lr \cdot \hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} \quad (2.16)$$

dimana:

t : iterasi ke- t , nilai awal $t = 0$

lr : *learning rate*

β_1 : *decay rate* untuk momentum, nilai umumnya adalah 0,9

β_2 : *decay rate* untuk gradien, nilai umumnya adalah 0,999

ϵ : konstanta 10^{-8}

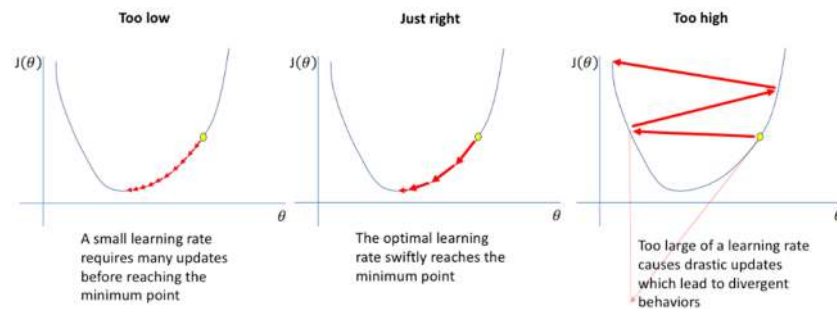
g_t : gradien saat iterasi ke- t

m_t : estimasi momen pertama

v_t : estimasi momen kedua

3. *Learning Rate*

Learning rate (lr) adalah salah satu parameter dalam pengoptimal yang mengontrol seberapa besar langkah yang diambil dalam penyesuaian bobot model. Pemilihan nilai *learning rate* sangat mempengaruhi hasil model karena ketika nilai *learning rate* terlalu besar, maka akan melewati nilai *loss* yang minimum sedangkan ketika nilai *learning rate* terlalu kecil, maka akan membutuhkan waktu yang sangat banyak dalam menuju titik konvergensi dan mendapatkan nilai *loss* minimum (Jordan, 2018).



Gambar 2.7 Perbandingan pemilihan *learning rate* (Jordan, 2018)

2.9 Metrik *Accuracy*

Evaluasi dilakukan dengan menguji model menggunakan dataset *test* dengan menghitung banyak prediksi yang benar dibagi dengan seluruh prediksi.

$$Acc = \frac{True\ Positive + True\ Negative}{All\ Prediction} \quad (2.17)$$

2.10 TensorFlow

TensorFlow adalah suatu *end-to-end platform* yang dibuat ramah untuk pemula untuk membuat model *Machine Learning* baik pada *platform desktop, mobile, website*, maupun *cloud* (TensorFlow Developer, 2023). Kenton (2022) menyatakan bahwa *end-to-end platform* berarti *platform* atau program yang menyediakan alat untuk memproses sesuatu dari awal hingga akhir dan memberikan solusi fungsional yang lengkap tanpa menggunakan pihak ketiga.

2.11 Matplotlib

Matplotlib adalah salah satu *library* untuk melakukan visualisasi data menggunakan python (Tineges & Davita, 2021). Pada penelitian ini, Matplotlib digunakan untuk memvisualisasikan kesalahan akurasi yang didapat oleh model.

2.12 Google Colaboratory

Google Colaboratory, yang juga dikenal sebagai *Google Colab*, merupakan sebuah platform *cloud* yang diberikan oleh Google untuk tujuan pembuatan, eksekusi, dan pembagian *notebook* interaktif yang menggunakan bahasa pemrograman *Python*. Di dalam lingkungan *Colab*, pengguna diberi kemampuan untuk menggabungkan kode, teks, serta elemen visual dalam sebuah dokumen tunggal yang dapat dijalankan langkah demi langkah. Hal ini memberikan kenyamanan bagi pengguna dalam melakukan analisis data, percobaan pada pembelajaran mesin, dan beragam tugas pemrograman lainnya tanpa keharusan untuk menyusun pengaturan lingkungan lokal atau perangkat keras.

BAB III

OBJEK DAN METODE PENELITIAN

3.1 Objek Penelitian

Objek pada penelitian ini adalah dataset gambar spesies bunga Edelweis yang terdiri dari data *train* dan data *test*. Secara berurutan, data untuk *training* dan data untuk *test* terdiri dari 3500 dan 1050 gambar yang tersebar ke dalam tiga buah kelompok sesuai spesiesnya yaitu *Anaphalis javanica*, *Leontopodium alpinum*, dan *leucogenes grandiceps*. Persebaran data *training* dan data *test* secara berturut-turut sebanyak 1666 dan 350 per spesies. Data yang akan digunakan dalam penelitian ini merupakan data sekunder yang diperoleh dari *website* Kaggle (Malau, 2022).

3.2 Metode Penelitian

Metode yang digunakan dalam penelitian ini adalah *Convolutional Neural Network* dan menggunakan algoritma Adam sebagai pengoptimal untuk klasifikasi spesies Edelweis menggunakan gambar dengan bahasa pemrograman *Python*. Pada penelitian ini, data akan diklasifikasikan ke dalam tiga kelas sesuai dengan jenis spesies pada dataset. Secara garis besar, langkah-langkah dalam penelitian ini terbagi ke dalam beberapa tahap dengan rincian sebagai berikut.

1. Pra-pemrosesan gambar

Pra-pemrosesan gambar dilakukan dengan melakukan kompresi gambar dengan mengecilkan ukuran *pixel* menjadi lebar sebesar 512px dan tinggi menyesuaikan dengan rasio tiap gambar serta mengurangi kualitas

gambar menjadi 85%. Kompresi dilakukan dengan tujuan untuk mempercepat proses komputasi dan menyeragamkan ukuran gambar. Kemudian pra-pemrosesan gambar akan dilanjutkan dengan membuat *image data generator*. Pada tahap ini, komponen warna pada gambar akan diubah menjadi angka dan dinormalisasi.

2. Pemisahan data *train* dan *validation*

Sebelum menggunakan data untuk membentuk model, data akan dibagi menjadi data *train* dan data *validation*. Sesuai dengan prinsip *10-fold cross validation*, rasio perbandingan antara data *train* dan data *validation* sebesar 8:2.

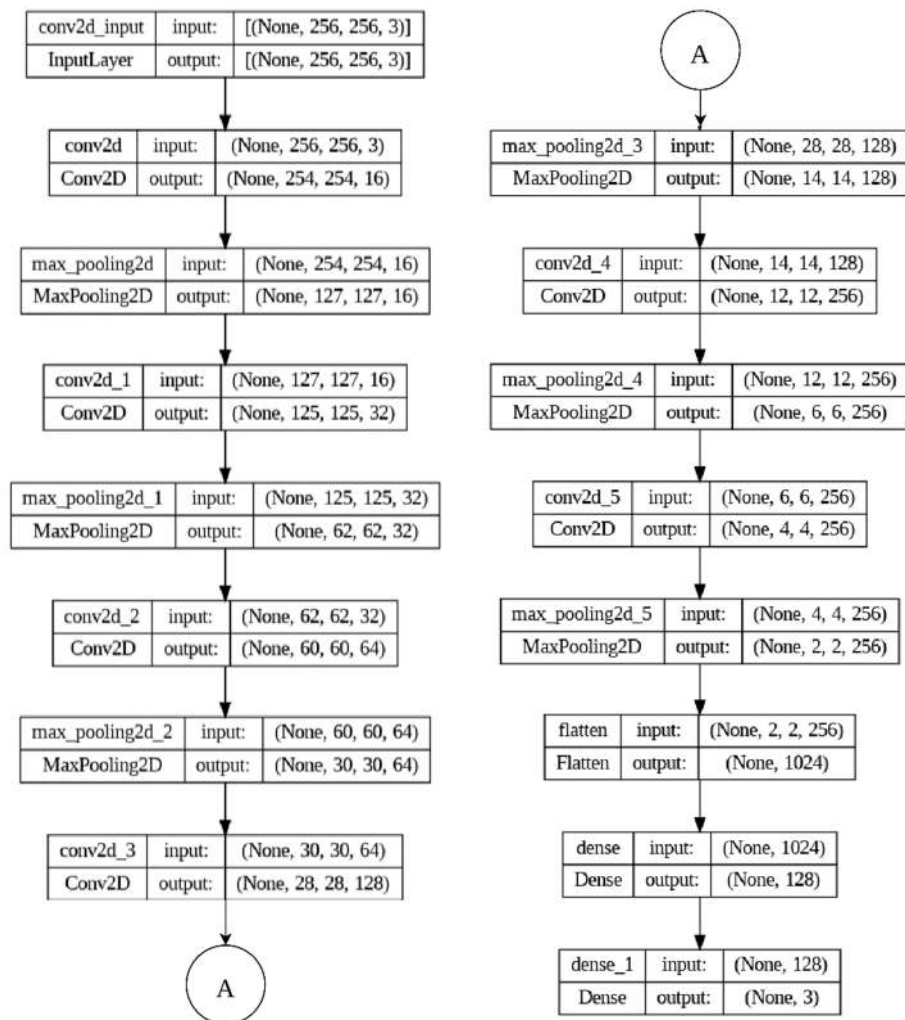
3. Inisiasi nilai learning rate sebesar 10^{-i} , dengan $i = 1$

4. *Training Model*

Tahap pertama dalam proses *training* model adalah membuat arsitektur model. Arsitektur model yang akan dibuat memiliki dua jenis tahapan yaitu tahap konvolusi yang terdiri dari *convolution layer* dengan operasi (2.2) dan *pooling layer*. Tahap selanjutnya adalah tahap klasifikasi yang diawali dengan mengubah dimensi *feature map* menjadi satu dimensi menggunakan *flatten layer*. Setelah menjadi satu dimensi, dilakukan proses klasifikasi menggunakan *fully-connected layer* berupa *dense layer* dengan aktivasi ReLU (2.4) dan *dense layer* dengan aktivasi *Softmax* (2.8).

Setelah pembuatan arsitektur model, dilakukan proses *training* model menggunakan data *train*. Langkah pertama pada proses *training* adalah dengan melakukan proses ekstraksi fitur menggunakan enam tahap

konvolusi dengan masing-masing banyak *filter* pada *convolution layer* di tiap tahap adalah 16, 32, 64, 128, 256, dan 256 filter dengan aktivasi ReLU.



Gambar 3.1 Arsitektur model CNN

Proses *training* model dilanjutkan dengan tahap klasifikasi. *Feature map* dari hasil tahap konvolusi terakhir diubah menjadi satu dimensi dan kemudian menjadi input untuk *dense layer* pertama yang terdiri dari 128 unit *neuron* dengan aktivasi ReLU (2.4). Output dari *dense layer* pertama kemudian menjadi input untuk *dense layer* terakhir dengan tiga unit *neuron*

sesuai dengan banyak kelas data dan menggunakan aktivasi *softmax*. Proses ini dilakukan secara berulang hingga 20 *epoch* dimana proses validasi model menggunakan data *validation* dan penyimpanan model dengan nilai bobot dan bias tertentu disimpan di setiap akhir *epoch*.

5. *Testing* model.

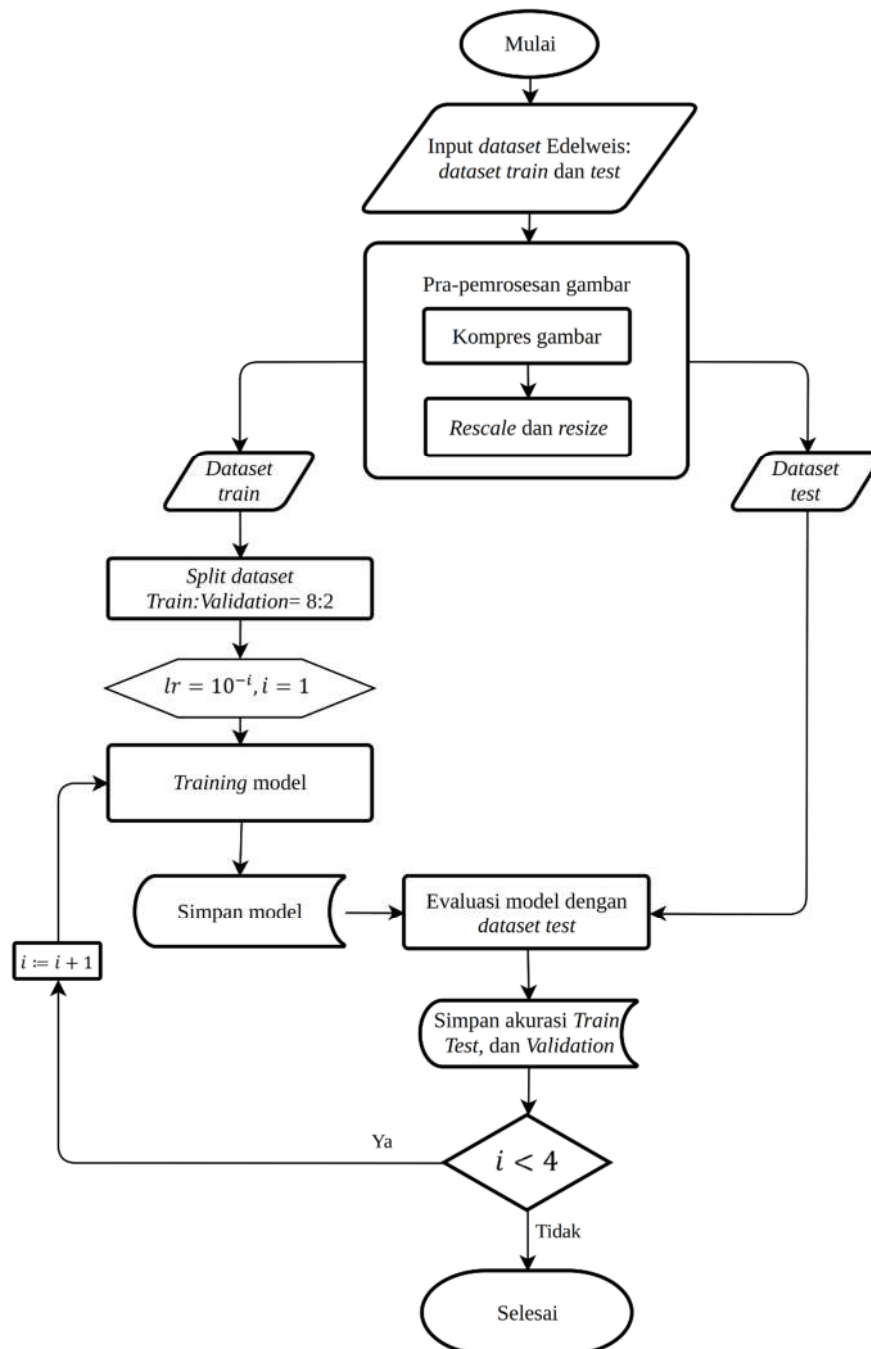
Setelah diperoleh model terbaik dari proses *training* sebanyak 20 *epoch*. Model yang disimpan pada *epoch* terakhir kemudian diuji menggunakan dataset *test*. Dataset test akan dimasukkan ke dalam model untuk dilakukan proses klasifikasi dan hasilnya akan disimpan.

6. Perbarui nilai i pada *learning rate* sehingga $i = i + 1$.

7. Ketika $i > 4$, proses *training* model berhenti.

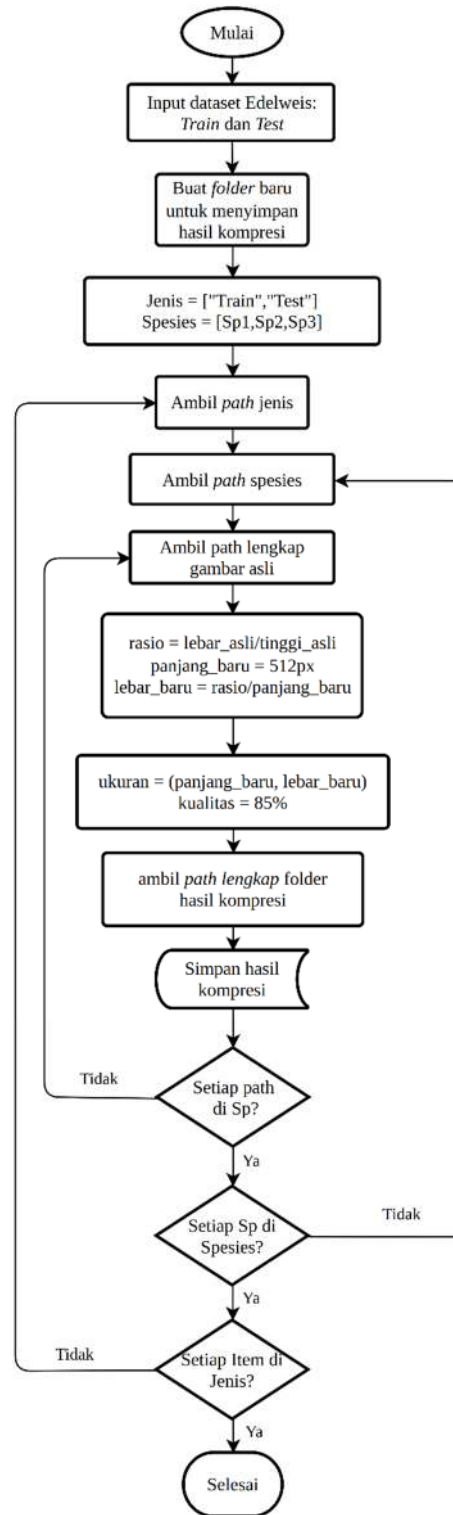
3.3 Diagram Alir Penelitian

Diagram alir penelitian klasifikasi spesies bunga Edelweis menggunakan *Convolutional Neural Network* dan pengoptimal Adam terdapat pada Gambar 3.2.



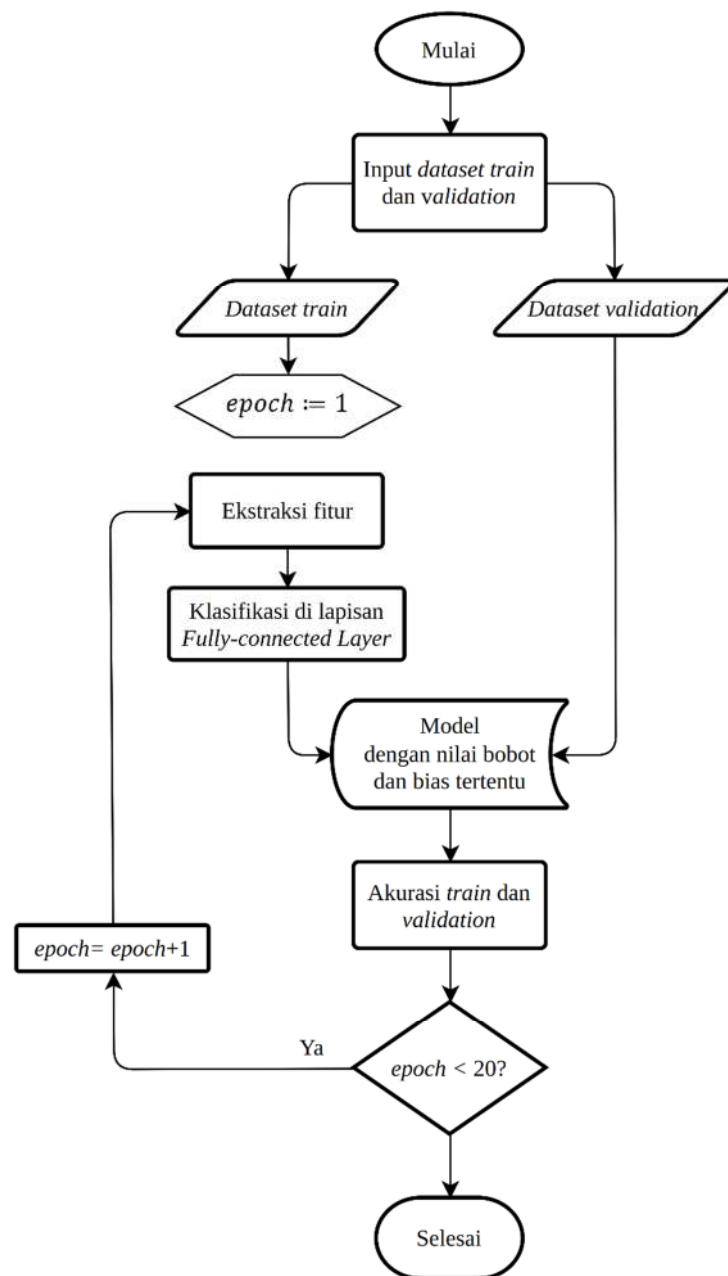
Gambar 3.2 Diagram alir penelitian

Diagram alir untuk tahap kompresi gambar dapat dilihat pada Gambar 3.3.



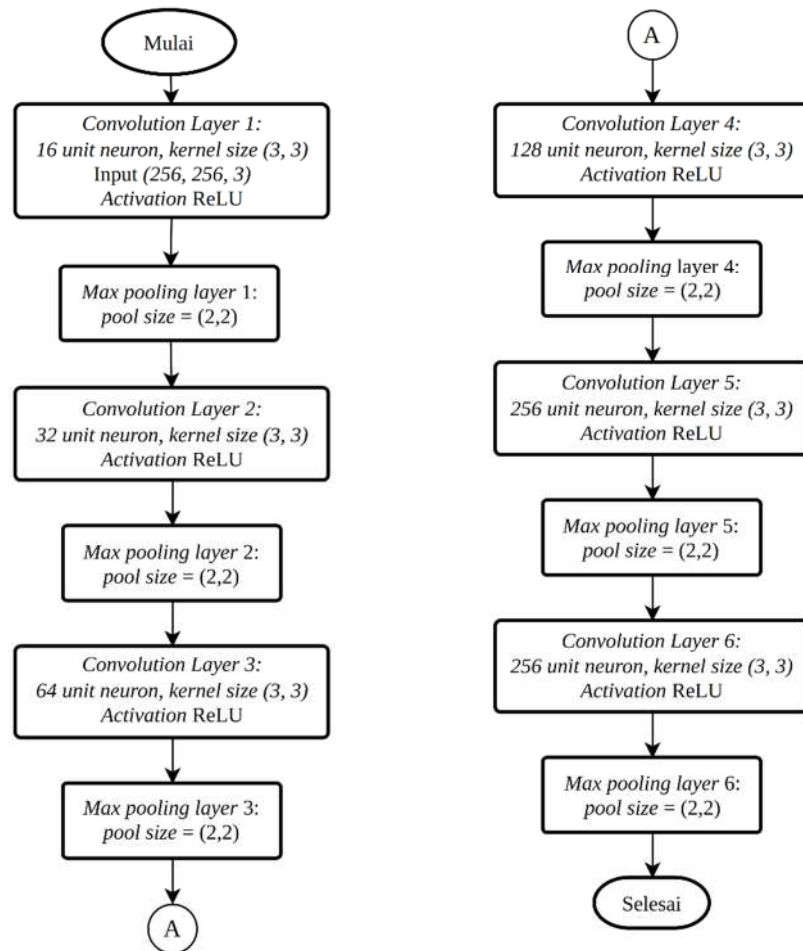
Gambar 3.3 Diagram alir tahap kompresi gambar

Diagram alir pada tahap *training* model dapat dilihat pada Gambar 3.4.



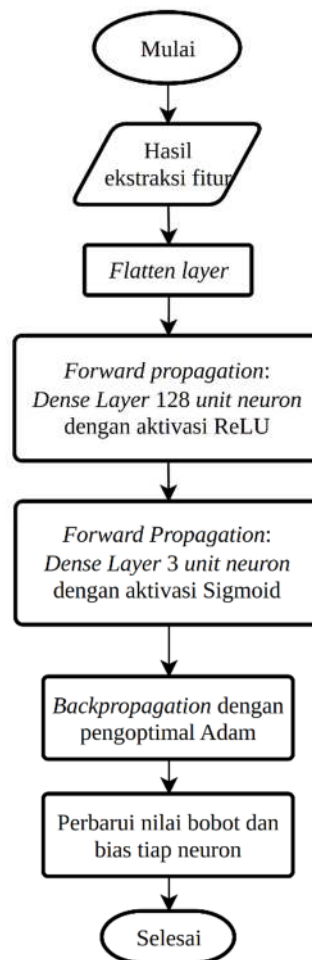
Gambar 3.4 Diagram alir pada tahap *training* model

Diagram alir pada tahap ekstraksi fitur dapat dilihat pada Gambar 3.5.



Gambar 3.5 Diagram alir pada tahap ekstraksi fitur

Diagram alir pada tahap klasifikasi di lapisan *Fully-connected Layer* dapat dilihat pada Gambar 3.6.



Gambar 3.6 Diagram alir pada tahap klasifikasi di lapisan *Fully-connected Layer*

DAFTAR PUSTAKA

- Alkaff, A. K., & Prasetyo, B. (2022). Hyperparameter Optimization on CNN Using Hyperband on Tomato Leaf Disease Classification. *2022 IEEE International Conference on Cybernetics and Computational Intelligence (CyberneticsCom)*, 479–483.
<https://doi.org/10.1109/CyberneticsCom55287.2022.9865317>
- Chaki, J., & Dey, N. (2018). *A beginner's guide to image preprocessing techniques*. CRC Press.
- Chauchan, N. S. (2020, December 18). *Optimization Algorithms in Neural Networks*. KDnugget. <https://www.kdnuggets.com/2020/12/optimization-algorithms-neural-networks.html>
- Géron, A. (2019). *Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow* (N. Tache, Ed.; 2nd ed.). O'Reilly Media, Inc.
<https://www.oreilly.com/library/view/hands-on-machine-learning/9781492032632/>
- Jordan, J. (2018, March 1). *Setting the learning rate of your neural network*.
<https://www.jeremyjordan.me/nn-learning-rate/>
- Kenton, W. (2022, May 24). *What Is End-To-End? A Full Process, From Start to Finish*. <https://www.investopedia.com/terms/e/end-to-end.asp>
- Ketkar, N. (2017). Convolutional Neural Networks. In *Deep Learning with Python: A Hands-on Introduction* (pp. 63–78). Apress.
https://doi.org/10.1007/978-1-4842-2766-4_5
- Kingma, D. P., & Ba, J. L. (2017). Adam: A Method for Stochastic Optimization. *The 3rd International Conference for Learning Representations*.
- LeCun, Y., Kavukcuoglu, K., & Farabet, C. (2010). Convolutional networks and applications in vision. *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*, 253–256.
<https://doi.org/10.1109/ISCAS.2010.5537907>
- Malau, F. R. (2022). *Edelweiss Flower Dataset*. Kaggle.
<https://www.kaggle.com/datasets/ndomalau/edelweis-flower>

- Malau, F. R., & Mulyana, D. I. (2022). Classification of Edelweiss Flowers Using Data Augmentation and Linear Discriminant Analysis Methods. *Journal of Applied Engineering and Technological Science*, 4(1), 139–148.
<https://doi.org/10.37385/jaets.v4i1.960>
- Muhammad, S., & Wibowo, A. T. (2021). Klasifikasi Tanaman Aglaonema Berdasarkan Citra Daun Menggunakan Metode Convolutional Neural Network (CNN). *EProceeding of Engineering*, 10621–10636.
- Shinozuka, M., & Mansouri, B. (2009). 4 - Synthetic aperture radar and remote sensing technologies for structural health monitoring of civil infrastructure systems. In V. M. Karbhari & F. Ansari (Eds.), *Structural Health Monitoring of Civil Infrastructure Systems* (pp. 113–151). Woodhead Publishing.
<https://doi.org/10.1533/9781845696825.1.114>
- TensorFlow Developer. (2023). *Introduction to TensorFlow*.
<https://www.tensorflow.org/learn>
- TensorFlow Developer. (2024a). *Conv2D*.
https://www.tensorflow.org/api_docs/python/tf/keras/layers/Conv2D
- TensorFlow Developer. (2024b). *Preprocessing Image: ImageDataGenerator*.
https://www.tensorflow.org/api_docs/python/tf/keras/preprocessing/image/ImageDataGenerator
- Tineges, R., & Davita, A. W. (2021). *Mengenal Matplotlib untuk Visualisasi Data dengan Python*. <https://dqlab.id/mengenal-matplotlib-untuk-visualisasi-data-dengan-python>

RIWAYAT HIDUP PENULIS