KLASIFIKASI JENIS PENYAKIT PADA CITRA DAUN PADI MENGGUNAKAN ALGORITMA CONVOLUTION NEURAL NETWORK

Dwi Nurani[1)] Imsak Lukiwidura Yanuar[2)] Andriyan Dwi Putra[3)]
Program Studi Informatika[1) 2)] Program Studi Sistem Informasi[3)]
Fakultas Ilmu Komputer, Universitas Amikom Yogyakarta, Yogyakarta[1) 2) 3)]
dwinurani@amikom.ac.id[1)] imsak.yanuar@students.amikom.ac.id[2)] andriyan.putra@amikom.ac.id[3)]

ABSTRACT

Rice (Oryza sativa) is one of the most important cultivated plants in civilization in the world. Including in Asia, making rice production as a staple food every day, especially in Indonesia. The quality and quantity of rice are certainly produced from healthy rice plants. However, there are many factors that can interfere with good rice production, one of which is disease in rice plants. Diseases in rice plants if not given the right treatment, the plants can wither and even die before harvesting. Rice plants that are infected with diseases that last until harvest time also cannot produce rice of good quality.

Rice farmers certainly have a warehouse for storing rice plants from the harvest. Therefore, different rice varieties are very easy to mix during harvesting, storage and marketing. Most rice farmers sort rice manually and do not pay much attention to plant health which will certainly require higher costs, subjectivity, boredom and inconsistencies associated with manual sorting.

Based on the results of other previous studies using methods other than CNN such as Transfer Learning, it obtained an accuracy of 92.46% and SVM (Support Vector Machine) with an accuracy of 82%. It is hoped that with CNN there will be an increase in accuracy.

Keyword: *rice, digital image, convolutional neural network algorithm, deep learning*

ABSTRAK

Tanaman padi (Oryza sativa) merupakan salah satu tanaman budidaya terpenting dalam peradaban di dunia. Termasuk di Asia, menjadikan hasil produksi padi sebagai bahan makanan pokok sehari hari, terutama di Indonesia. Kualitas dan kuantitas beras tentu dihasilkan dari tanaman padi yang sehat. Namun, ada banyak faktor yang dapat mengganggu produksi beras yang baik, salah satunya adalah penyakit pada tanaman padi. Penyakit pada tanaman padi jika tidak diberikan penanganan yang benar, maka tanaman dapat layu bahkan mati sebelum dipanen. Tanaman padi yang terjangkit penyakit yang bertahan hingga masa panen juga tidak bisa menghasilkan beras dengan kualitas yang baik.

Para petani padi tentu memiliki Gudang penyimpanan tanaman padi dari hasil panen. Oleh karena itu varietas padi yang berbeda – beda sangat mudah tercampur selama panen, penyimpanan, dan pemasaran. Kebanyakan para petani padi mengurutkan padi secara manual dan tidak begitu memperhatikan Kesehatan tanaman yang tentu akan membutuhkan biaya tinggi, subjektivitas, kebosanan dan inkonsistensi terkait dengan pengurutan manual tersebut.

Berdasarkan hasil penelitian lain sebelumnya menggunakan metode selain CNN seperti Transfer Learning mendapatkan akurasi 92,46% dan SVM (Support Vector Machine) dengan akurasi 82%. Diharapakan dengan CNN akan terjadi peningkatan akurasi.

Kata Kunci: *tanaman padi*, citra digital, Algoritma *Convolutional Neural Network, deep learning*

PENDAHULUAN

Padi (Oryza sativa) tanaman pokok yang menjadi komoditas penting bagi masyarakat Asia, terutama Indonesia. Masyarakat Indonesia yang setiap hari untuk memenuhi kebutuhan pangan dari semua kalangan tidak jauh dari beras yang dihasilkan oleh tanaman padi. Menurut data dari BPS (Badan Pusat Statistik) tahun 2020 luas panen diperkirakanmencapai 10,79 juta hektar mengalami kenaikan sebanyak 108,93 ribu hektar atau 1,02 persen dibadingkan luas panen pada tahun 2019 yang hanya 10,68 juta hektar [1]. Diharapkan dengan adanya peningkatan luas panen padi produksi beras juga ikut meningkat di Indonesia. Tetapi, tentu ada banyak permasalahan yang harus dihadapi oleh para petani padi yang mempengaruhi hasil panen padi, beberapa di antaranya seperti masalah hama, cuaca dan penyakit pada tanaman padi tersebut. Salah satu permasalahan yang sulit ditangani karena terbatasnya pengetahuan petani di Indonesia adalah penyakit pada tanaman padi. Terbukti dengan banyaknya petani yang melakukan kesalahan dalam cara pengobatan padi karena kesalahan dalam mengidentifikasi penyakit pada tanaman padi tersebut [2].

Beberapa faktor penyakit pada tanaman padi disebabkan seperti bakteri, jamur, nematoda (biotik) dan faktor lingkungan berupa suhu, cuaca, dan tingkat kelembapan (abiotik). Tanaman padi yang terjangkit penyakit mempengaruhi kualitas dan kuantitas beras yang akan dihasilkan. Tanaman padi yang terkena penyakit tidak dapat melakukan proses fotosintesis dengan maksimal sehingga mempengaruhi kuantitas dan kualiatas beras menjadi turun [3].

Para petani padi tentu memiliki Gudang penyimpanan tanaman padi dari hasil panen. Oleh karena itu varietas padi yang berbeda – beda sangat mudah tercampur selama panen, penyimpanan, dan pemasaran. Kebanyakan para petani padi mengurutkan padi secara manual dan tidak begitu memperhatikan Kesehatan tanaman yang tentu akan membutuhkan biaya tinggi, subjektivitas, kebosanan dan inkonsistensi terkait dengan pengurutan manual tersebut.

Melihat permasalahan yang ada peneliti berinisiatif untuk melakukan penelitian mengenai klasifikasi tanaman padi menggunakan algoritma Conolutional Neural Network serta menyelidiki peforma kinerja algoritma CNN dalam klasifikasi varietas penyakit daun padi yang berasal dari dataset Rice Leaf Diseases Dataset. Kelas data yang digunakan sejumlah 3 kelas dari 120 pada dataset Rice Leaf Diseases. Pada penelitian [4] dengan menggunakan *Transfer Learning*, sistem dapat mengklasifikasi tiga kelas penyakit tanaman padi yaitu busuk batang, hawar daun bakteri, dan bercak daun dengan akurasi sebesar 92,46%. Pada penelitian [5] dengan menggunakan klasifikasi SVM (Support Vector Machine) didapatkan hasil akurasi sebsar 82% untuk identifikasi penyakit busuk batang.

TINJAUN PUSTAKA

Analisis

Menurut kamus besar Bahasa Indonesia, analisis adalah penelitian terhadap suatu peristiwa / kejadian seperti karangan, perbuatan, dan lain sebagainya untuk mengetahui kebenaran dari suatu peristiwa / kejadian (sebab-akibat, duduk prakarya, dan sebagainya).

Secara umum analisis adalah sebuah kegiatan menguraikan atau mencari bagian-bagian kecil seperti kepingan puzzle dari suatu masalah sampai ditemukan hubungan antara tiap bagian.

Citra Digital

Citra digital merupakan representative dari citra yang diambil oleh mesin dalam bentuk pendekatan berdasarkan sampling dan kuantisasi. Sampling menyatakan besar tiap kotak yang disusun dalam baris dan kolom. Dengan asumsi lain, sampling yang terdapat pada citra menyatakan besar kecilnya ukuran *pixel* (titik) pada citra, kuantitasi menyatakan besar nilai tingkat kecerahan yang dinyatakan dalam nilai tingkat keabuan (*grayscale)* sesuai dengan jumlah bit biner yang digunakan mesin, dengan kata lain kuantisasi pada citra menyatakan jumlah warna yang ada pada citra [7].

Dalam mesin computer, citra digital dipetakan dalam bentuk grid dan elemen *pixel* berbentuk

matriks 2 dimensi. Setiap piksel tersebut memiliki angka yang mempresentasikan channel warna. Angka setiap pixel disimpan berurutan oleh computer dan sering sekali dikurangi untuk keperluan kompresi atau pengolahan tertentu. Suatu citra digital dapat mewakili oleh sebuah matriks yang terdiri dari M kolom dan N baris, dimana perpotongan antara kolom dan baris disebut piksel. (*Pixel* = picture element) yaitu elemen kecil dari citra. Piksel memiliki dua parameter, yaitu koordinat dan intensitas atau warna. Nilai terdapat pada koordinat (x,y) adalah f(x,y) yaitu besar intensitas atau warna dari piksel pada titik tersebut. Karena itu, citra dapat dituliskan dalam bentuk matriks :

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \cdots & f(0,M-1) \\ f(1,0) & f(1,1) & \cdots & f(1,M-1) \\ \vdots & \vdots & \vdots & \vdots \\ f(N-1,0) & f(N-1,1) & \cdots & f(N-1,M-1) \end{bmatrix}$$

Gambar 1 Matriks Konvolutional

Berdasarkan rumus diatas, suatu citra f(x,y) dapat dituliskan dalam fungsi matematis seperti berikut ini :

$$0 \le x \le M-1$$
$$0 \le x \le N-1$$
$$0 \le f(x,y) \le G-1$$

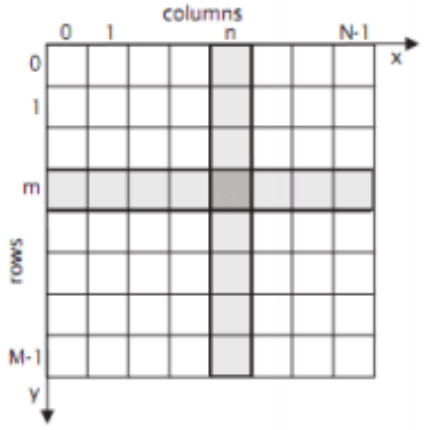
Diketahui

M = jumlah piksel baris pada array citra
N = jumlah piksel kolom pada array citra
G = nilai skala keabuan (*grayscale*)

Besarnya nilai M, N, dan G biasanya merupakan perpangkatan dari dua seperti yang terlihat pada persamaaan berikut ini :

M = 2m ; N = 2n ; G = 2k

Dimana nilai **m, n,** dan **k** merupakan bilangan positif. Interval (0,G) disebut dengan istilah (*grayscale*). Jumlah nilai G tergantung proses digitalisasinya. Keabuan 0 (nol) menyatakan intensitas hitam sedangkan 1 (satu) menyatakan intensitas putih. Untuk citra 8 bit. Nilai untuk G sama dengan 28 = 256 warna (derajat keabuan)



Gambar 2 Representasi Citra Digital dalam 2 Dimensi

Supervised Learning

Supervised Learning menjadi salah satu skenario dalam pembelajaran *Machine Learning*. Pemanfaatan skenario *Supervised Learning* seperti pembelajaran menggunakan masukan data pembelajaran yang sudah diberi label. Kemudian membuat prediksi dari data tersebut. Contoh Decision Tree, Nearest-Neighbor Classifier, Naïve Bayes Classifier, Artificial Neural Network, Support Vector Machine, Fuzzy K-Nearest Neighbor [11].

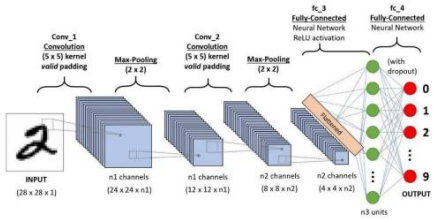
Deep Learning

Deep learning merupakan salah satu bidang dalam Machine Learning yang memanfaatkan jaringan syaraf tiruan untuk implementasi permasalahan dengan dataset besar. Teknik deep learning memberikan arsitektur kuat untuk supervised learning. Menambahkan lapisan lebih, maka model pembelajaran tersebut bisa mewakili data citra berlabel dengan baik. Deep learning memungkinkan model komputasi yang terdiri dari beberapa lapisan pengolahan untuk representasi data dengan beberapa tingkat abstraksi. Deep learning bisa diartikan sebagai sebuah model yang mempelajari komputasi dengan kemampuan "otak sendiri" dan bisa mengambil keputusan dengan meniru logika manusia yang dibuat.

Convolutional Neural Network

Convolutional Neural Network (CNN) adalah salah satu metode yang biasa digunakan dalam kasus image classifier. CNN bisa disebut sebagai metode deep learning karena menggunakan banyak layer yang terdiri dari Convolutional Layer dan Pooling Layer sebagai lapisan Feature Learning. Selain itu pada lapisan Classification terdapat Flatten sebagai input dari Fully-Connected Layer dan Fully-Connected Layer sebagai penghubung setiap neuron dalam lapisan yang digunakan untuk menghitung skor kelas.

Dengan demikian metode CNN bisa diartikan metode untuk transformasi gambar original layer tiap layer dari nilai pixel gambar ke dalam nilai skor kelas untuk mengklasifikasi serta setiap layer ada yang memiliki hyperparameter dan begitu sebaliknya, berikut gambar penjelasannya :

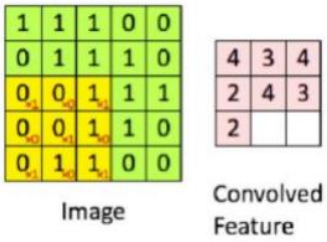


Gambar 3 Arsitektur CNN
(Towards Data Science Sumit Saha, 2018)

Berdasarkan pada gambar 3 ada tahap awal/pertama pada arsitektur CNN yaitu konvolusi pada convolutional layer, *Convolutional layer* melakukan konvolusi output dari layer sebelumnya. Perhitungan konvolusi dengan sebuah kernel memiliki ukuran tertentu. Kemudia dilanjutkan dengan fungsi aktivasi yaitu ReLU (Rectifier Linear Unit) setelah itu masu pada proses *pooling* pada *pooling layer.* Tahap-tahap tersebut diulang hingga mendapatkan *feature map* yang cukup untuk dilanjutkan pada tahapi *fully-connected.*

1. Convolutional Layer (Layer Konvolusi)

Layer convolution merupakan layer pertama pada arsitektur CNN. Menurut penelitian [12] bahwa *Layer convolution* merupakan layer pertama yang menerima input gambar langsung pada arsitektur. Layer ini melakukan operasi konvolusi yaitu operasi kombinasi linier filter terhadap daerah lokal.



Gambar 4 Operasi konvolusi

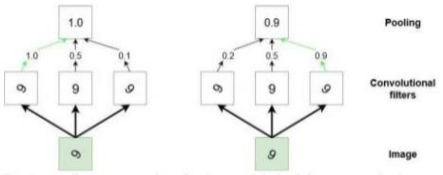
Perhitungan operasi konvolusi pada convolutional layer dengan mengalihkan indeks pada input image *f* dan indeks kernel *h* untuk mendapatkan feature map dan dapat dirumuskan seperti berikut :

$$G[m,n] = (f * h)[m,n] \quad (2.4)$$

G = Feature map
f = indeks input range
m = indeks baris
h = indeks kernel size
n = indeks kolom

2. Pooling layer

Pooling layer adalah layer yang biasa digunakan untuk mempercepat perhitungan, melalui proses pooling ukuran gambar akan lebih kecil. Pooling layer menurut penelitian [12] akan mereduksi ukuran spasial dan jumlah parameter jaringan serta mempercepat komputasi dan mengontrol *overfitting*. Adapun penelitian [13] mengatakan bahwa *pooling layer* adalah sebuah proses reduksi ukuran sebuah citra yang bertujuan untuk meningkatkan inviariansi posisi dari fitur. Penelitian ini menggunakan *pooling layer* yang digunakan adalah *Max-pooling. Max-pooling* sendiri mencari nilai maksimal dalam iterasi dan mengambil satu nilai terbesar. Berikut contoh gambar *Max-pooling* pada *pooling layer*.



Gambar 5 *Pooling Layer*
(PDF) Using the Choquet Integral in the Pooling Layer in Deep Learning Networks (researchgate.net)

3. Fully-Connected Layer

Fully-connected layer merupakan layer yang menghubungkan setiap neuron dalam lapisan yang digunakan untuk menghitung skor kelas. Input fully-connected adalah berupa vector sehingga sebelum melakukan fully-connected akan melakukan flatten terlebih dahulu. Hasil dari feature map masih berbentuk multidimensional array. Array pada data pooling dikonversikan menjadi data satu dimensi single vector.

Epoch

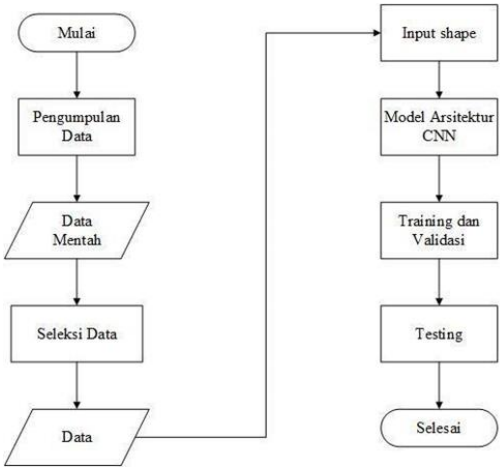
Epoch adalah hyperparameter yang menentukan berapa kali algoritma pembelajaran akan bekerja mengolah data training keseluruhan. Jumlah epoch adalah jumlah complete pass yang harus dilewati pada dataset training. Jumlah epoch biasanya besar, sering ditemukan ratusan hingga ribuan dengan tujuan meminimalkan kesalahan model ketika sedang berjalan melakukan algoritma pembelajaran. Kita dapat melihat contoh jumlah *epoch* dari banyak literature ke 10, 100, 500, 1000, dan lebih besar lagi.

Learning Rate

Learning Rate adalah salah satu dari parameter training untuk menghitung nilai koresksi berat pada waktu proses training dilakukan. Untuk nilai learning rate itu sendiri berada pada range nol (0) sampai (1). Semakin besar nilai learning rate, maka semakin cepat proses training akan berjalan. Apabila learning rate semakin besar, maka untuk masalah ketelitian jaringan akan semakin berkurang dan berlaku begitu *sebaliknya. Jilak learning* rate semakin kecil, ketelitian semakin besar namun dengan proses training yang akan memakan waktu lebih lama.

METODE PENELITIAN

Tahap Penelitian



Gambar 6 Alur Penelitian

Metode Pengumpulan Data

Metode pengumpulan data yang akan dilakukan untuk penelitian ini menggunakan dataset gambar daun tanaman padi yang didapatkan dari situs (https://www.kaggle.com/vbookshelf/rice-leaf-diseases) menggunakan data image Rice Leaf Diseases Dataset,serta metode pengumpulan data yang digunakan dalam penelitian berupa studi literatur dan Pustaka. Peneliti mengumpulkan data yang bersumber dari internet, e-book, atau jurnal online yang berkaitan dengan penelitian sebagai referensi untuk menyelesaikan tugas akhir ini.

Metode Analisis

Peneliti melakukan beberapa tahap analisis dengan sebagai berikut:

1. Menganalisa tahapan demi tahapan pada proses pembuatan model CNN hingga menemukan model yang tepat
2. Melakukan processing dengan melakukan penyamaan ukuran pada gambar, resize dilakukan saat augmentasi data sehingga ukuran data yang telah di augmentasi menjadi sama dengan ukuran data yang lain.

3. Menganalisis hasil nilai Loss dan Accuracy pada model yang telah dibuat
4.Menganalisis parameter yang mempengaruhi nilai Loss dan Accuracy pada proses klasifikasi

Metode Perancangan

Pada proses perancangan ini akan dilakukan berbagai tahapan pembuatan model Convolutional Neural Network dengan menggunakan model sequintal. Pembagian data untuk proses training dan validasi, pembagian tersebut melalui data training sebesar 80% dan sisanya untuk validasi. Proses training untuk melatih model CNN supaya memperoleh tingkat akurasi yang tinggi. Pada proses training dan validasi terdapat variable Loss, Accuracy (akurasi), Val_Loss (validation loss), dan Val_Acc (validationa accuracy). Beberapa library pendukung yang digunakan adalah library Tensorflow dan Keras.

Metode Implementasi

Metode implementasi dilakukan dengan melakukan uji coba data training dan validasi setelah melakukan pembagian yaitu 120 file gambar untuk training dan validasi. Pemograman yang digunakan yaitu *Python 3* dan menggunakan GPU notebook pada Google Colab.

Metode Testing

Tahap demi tahap pada penelitian sudah dilakukan dan mendapatkan nilai akurasi yang maksimal, maka akan dilakukan testing atau pengujian. Proses pengujian merupakan proses klasifikasi menggunakan bobot dari hasil proses training. Proses training yang dilakukan untuk mengklasifikasi citra jenis penyakit tanaman padi masing-masing. Adapun pada metode testing yang akan digunakan adalah proses feedforward. Proses feedforward yang telah dilakukan menghasilkan lapisan output.

Hardware

Hardware merupakan bagian fisik dari sebuah computer. Hardware pada komputer merupakan bagian penting yang menunjang pada penelitian. Spesifikasi dari computer yang digunakan pada penelitian ini dicantumkan pada tabel dibawah.

Tabel 1 Daftar Hardware

Hardware	
Processor	**Intel core i5-7200U**
RAM	**12GB**
GPU	**Intel(R) HD Graphics 620**
Motherboard Merk	**X456URK**

Software

Software yang digunakan yaitu :

1. Google Colab
2. Google Chrome
3. Microsoft Windows 10 Pro version 10.0.19043 Build 19043

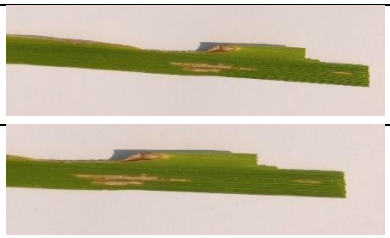
Data Mentah

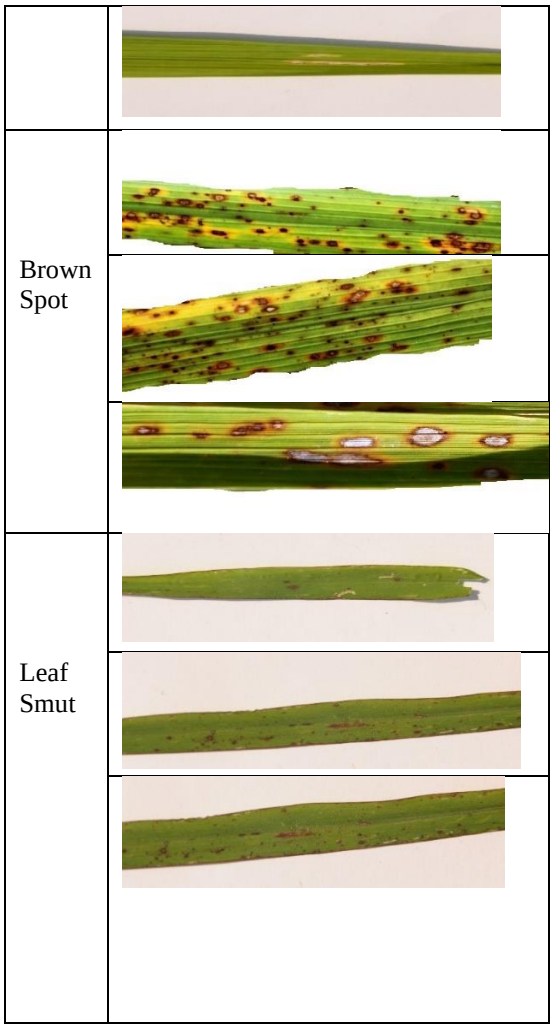
Dataset yang digunakan berasal dari kaggle berupa Rice Leaf Diseases Dataset dan 3 kelas citra penyakit tanaman padi pada dataset tersebut. Dataset tersebut yang akan dijadikan sebagai objek penelitian klasifikasi citra penyakit tanaman padi menggunakan CNN.

Seleksi Pembagian Data

Proses seleksi data memiliki tujuan untuk membuang data gambar yang tidak diperlukan pada penelitian ini. Pada dataset *Rice Leaf Diseases Dataset* terdapat 3 kelas jenis penyakit yang berisi data gambar setiap kelas jenis penyakit tanaman padi, sehingga dan tidak ada data lain. Jadi tidak diperlukan pembuangan dikarenakan semua data digunakan.

Tabel 2 Gambar Penyakit Padi

Jenis penyakit tanaman padi	Gambar
Bacterial Leaf Blight	



Setelah melakukan seleksi, Langkah berikutnya adalah melakukan pembagian data terhadap dataset berupa data training dan data validasi. Pembuatan untuk folder training dan validasi untuk setiap gambar jenis penyakit daun padi dilakukan oleh peneliti dengan persentase yaitu 80% untuk validasi dan 20% untuk training, dari hasil pembagian ditemukan data training dan validasi sebanyak 32 dan 8 data gambar untuk jenis penyakit *Bacterial Leaf Blight*, *Brown Spot* sebanyak 32 data training dan 8 data validasi, dan untuk jenis penyakit Leaf Smut sebanyak 32 data training dan 8 data validasi.

Tabel 3 Pembagian dataset

DATA	Training	Validasi	Total
Bacterial Leaf Blight	**32**	**5**	**40**
Brown Spot	**32**	**8**	**40**
Leaf Smut	**32**	**8**	**40**
Total	**96**	**24**	**120**

Pada tabel diatas dapat dilihat keseluruhan data adalah 120 data, banyak data untuk *training* adalah 96 data gambar dari ketiga jenis penyakit daun padi sedangkan untuk validation 24.

Tahap Implementasi

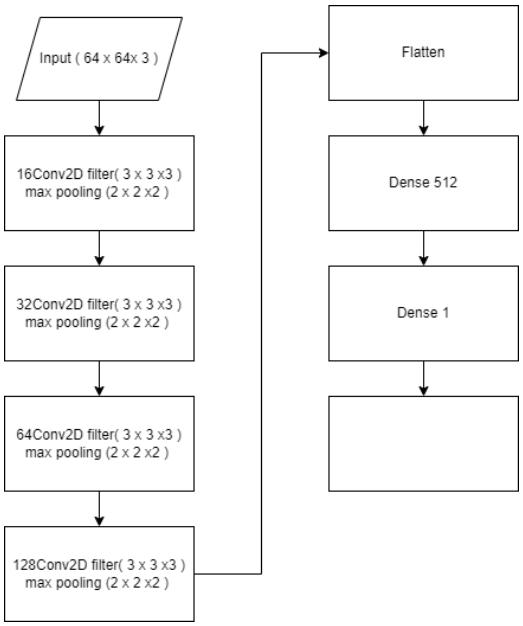
Tahap ini merupakan kegiatan untuk mengimplementasikan rancangan yang telah disusun agar dapat diwujudkan, yaitu dengan meletakkan sistem supaya siap dioperasikan. Rancangan yang telah disusun tentunya dibuat sesuai dengan prosedurnya supaya sistem dapat dioperasikan.

Input Shape

Setelah tahap pengumpulan data dan pembagian data, maka tahap berikutnya menentukan input shape tahap ini juga biasa disebut *processing*. Ukuran pada semua data akan diubah dengan di *Resize* untuk seluruh gambar mempunyai ukuran yang sama dan untuk memudahkan selama proses *training*. Keseluruhan ukuran menjadi 64 x 64 pixel pada data gambar.

Training dan Validasi

Pada tahap setelah melakukan input shape, Langkah berikutnya yaitu merancang model untuk arsitektur *Convoloutional Neural Network.* Alur pembuatan model CNN seperti diagram berikut.



Gambar 8 Model Arsitektur CNN

Pada diagram di atas proses learning pertama yaitu menentukan input shape yang dimana pada penelitian ini menggunakan ukuran 64x64 piksel, semua gambar diubah sesuai input shape yang sudah ditentukan, setelah itu proses konvolusi pertama yaitu menggunakan filter 16 dan kernel dengan matriks 3x3. Kemudian dilakukan proses *pooling* menggunakan ukuran *pooling* 2x2. Proses pooling dilakukan untuk mengurangi ukuran spasial representasi dan mengurangi banyaknya parameter juga perhitungan pada arsitektur jaringan CNN, proses pooling pada penelitian ini menggunakan *Max Pooling* karena metode ini mengambil salah satu pixel terbesar pada luas pixel suatu gambar dengan demikian tidak mengurangi informasi gambar tersebut. kemudian konvolusi kedua dengan menggunakan jumlah filter sebanyak 32 dan kernel dengan matriks 3x3 lalu dilanjutkan dengan proses *pooling* ukuran yang sama pada konvolusi pertama dan berlanjut ke konvolusi ketiga yaitu dengan filter 64 dengan kernel yang sama. Kemudian dilanjutkan dengan proses *pooling* ukuran yang sama dengan konvolusi pertama dan berlanjut kek konvolusi keempat yaitu dengan filter 128 dengan kernel yang sama. Setelah proses konvolusi dilanjutkan dengan *flatten* yaitu mengubah output dari proses konvolusi yang berbentuk matriks menjadi sebuah vector yang akan diteruskan pada proses klasifikasi.

Berikut ini adalah mode summary dari model CNN yang telah dibuat seperti pada gambar di bawah:

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 62, 62, 16)        448

 max_pooling2d (MaxPooling2D  (None, 31, 31, 16)       0
 )

 conv2d_1 (Conv2D)           (None, 29, 29, 32)        4640

 max_pooling2d_1 (MaxPooling  (None, 14, 14, 32)       0
 2D)

 conv2d_2 (Conv2D)           (None, 12, 12, 64)        18496

 max_pooling2d_2 (MaxPooling  (None, 6, 6, 64)         0
 2D)

 conv2d_3 (Conv2D)           (None, 4, 4, 128)         73856

 max_pooling2d_3 (MaxPooling  (None, 2, 2, 128)        0
 2D)

 flatten (Flatten)           (None, 512)               0

 dense (Dense)               (None, 512)               262656

 dense_1 (Dense)             (None, 3)                 1539

=================================================================
Total params: 361,635
Trainable params: 361,635
Non-trainable params: 0
_________________________________________________________________
```

Gambar 9 Model Summary

Dari gambar diatas model CNN yang dibuat didapatkan total parameter sebanyak 300 ribu lebih parameter. Parameter sendiri adalah sesuatu yang dipelajari pada citra atau bisa dikatakan sebagai pegangan untuk mengenali suatu citra tersebut, maka suatu parameter dibutuhkan. Kemudian menuju proses konvolusi, konvolusi pertama memiliki parameter bernilai 448, tedapat weight matrik pada proses konvolusi sehingga terdapat parameter yang dipelajari pada proses konvolusi. Pada pooling layer (max pooling) tidak terdapat parameter dikarenakan pada

pooling layer hanya mengambil nilai terbesar *featuremap. Full-connected* mempunyai nilai parameter terbanyak disini. Dikarenakan setiap neuron terhubung ke neuron lainnya. Berikut adalah contoh perhitungan parameter pada *model summary*.

1. Param Conv 1 = 16 (8 (3 x 3) + 1 = 448
2. Param Conv 2 = 32 (16 (3 x 3) + 1 = 4640
3. Param Conv 3 = 64 (32 (3 x 3) + 1 = 18496
4. Param Conv 4 = 128 (64 (3 x 3) + 1 = 73856
5. Param Dense 1 = (73856 + 1) x 512 = 262656
6. Param Dense 2 = (512 x 1) + 3 = 1536

Testing

Setelah training sudah dilakukan, berikutnya dilakukan *testing* pda data uji yang sudah disiapkan. Para proses *Testing* ini terdapat proses *feedforward* yang terdapat di setiap layer dengan memprediksi gambar yang memiliki output berupa klasifikasi gambar. Semua gambar pada tahap *training* diubah atau dilakukan *resize* menjadi 64 x 64 pixel. Hasil dari pengujian atau prediksi adalah jika outputnya 0 maka gambar tersebut *Bacterial Leaf Blight*, jika outputnya 1 maka gambar tersebut adalah *Brown Spot*, sedangkan jika outputnya 2 maka gambar tersebut adalah jenis penyakit *Leaf Smut.*

IMPLEMENTASI

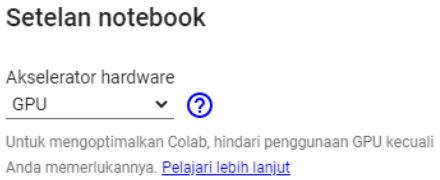
Berdasarkan identifikasi masalah yang sudah dibahas pada bab 1 telah ditemukan rumusan bagaimana cara implementasi klasifikasi citra penyakit tanaman padi menggunakan Convolutional Neural Network dan cara melakukan peningkatan terhadap tingkat akurasinya. Dengan rumusan masalah diatas maka peneliti membuat program dengan menggunakan Google Colab dan bahasa pemograman bahasa python.

Berikut tahap implementasi pada Google Colab

Inisialiasi Google Colab

Pada tahap ini dilakukan dengan menganalisa direktori penyimpanan pada dataset penelitian di *google drive*. Sebelum melakukan inisialisasi pada *google colab* diperlukan sinkronisasi ke *google drive* dimana dataset penelitian ini disimpan. Sinkornasi dilakukan dengan cara mengisi kode otorisasi pada prompt yang akan muncul pada google colab dengan menekan URL lalu masuk ke akun *google drive* dan menyalin kode otorisasi. Berikut ini tahapan sinkronasi untuk google colab ke google drive :

1. Sebelum masuk menuju google colab, dilakukan pengaturan terhadap notebook colab yaitu dengan melakukan pengubahan akselerator *Hardware* ke GPU. Untuk melakukan pengubahan bisa dengan pilih menu edit, lalu pilih notebook setting, lalu atur *hardware accelerator* ke GPU. Untuk mengoptimalkan Colab, hindari penggunaan GPU kecuali Anda memerlukannya.



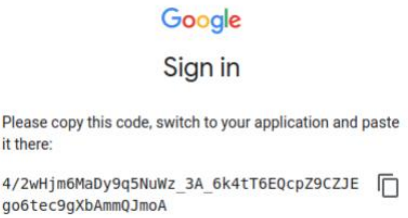
Gambar 10 Pengaturan Notebook

2. Kemudian untuk melakukan sinkronasi *colab ke drive* dilakukan dengan melakukan penulisan kode seperti yang ada dibawah ini, dengan demikian akan muncul URL dan input text untuk mengisi kode otorisasi.

```
from google.colab import drive
drive.mount('/content/drive')
```

Gambar 11 Otorisasi

3. Klik URL tersebut untuk melakukan sinkronasi akun google drive yang akan digunakan dan nantinya akan mendapatkan sebuah kode otorisasi lalu salin kemudian tempel pada input teks sebelumnya



Gambar 12 Kode Otorisasi

4. Setelah menyalin dan menempel kode otorisasi pada input teks kemudian tekan enter dan akan muncul “Mounted at /content/MyDrive” maka proses sinkronasi ke google drive telah berhasil dan selesai

```
[5] from google.colab import drive
    drive.mount('/content/drive')

    Mounted at /content/drive
```

Gambar 13 Setelah Melakukan Otorisasi

Inisialisasi Direktori Google Drive

Setelah menghubungkan menuju google drive, Langkah berikutnya yaitu menginisialisasi direktori penyimpanan dataset yang akan digunakan pada penelitian ini. Berikut ini *source code* untuk menginisialisasi direktori dan menampilkan banyaknya data training pada setiap jenis penyakit tanaman padi.

```
[3] import zipfile, os

    # Menginisialisasi direktori penyimpanan dataset
    zip_location = "/content/drive/MyDrive/Dataset_Skripsi/dataset beras.zip"

    from google.colab import drive
    drive.mount('/content/drive')

    Mounted at /content/drive

[8] # Membaca file zip menggunakan method ZipFile
    zip_read = zipfile.ZipFile(zip_location,'r')

    # Mengekstrak data yang ada pada file zip
    zip_read.extractall('/content/drive/MyDrive/Dataset_Skripsi')
```

Gambar 14 Inisialisasi Direktori Dataset

Setelah inisialisasi direktori utama kemudian akan dilakukan pembagian data training dan validasi menggunakan library *splitfolders* berikut ini adalah source code pembagian data dan inisisialisasi setiap direktori jenis penyakit tanaman padi, pembagian data training dan validiasi adalah sejumlah 80% dan 20%

```
[16] print("jumlah data training Bacterial leaf blight", len(os.listdir(Bacterial_leaf_blight_dir_train)))
     print("jumlah data training Brown spot", len(os.listdir(Brown_spot_dir_train)))
     print("jumlah data training Leaf smut", len(os.listdir(Leaf_smut_dir_train)))

     jumlah data training Bacterial leaf blight 32
     jumlah data training Brown spot 32
     jumlah data training Leaf smut 32
```

Gambar 15 Total Data Training

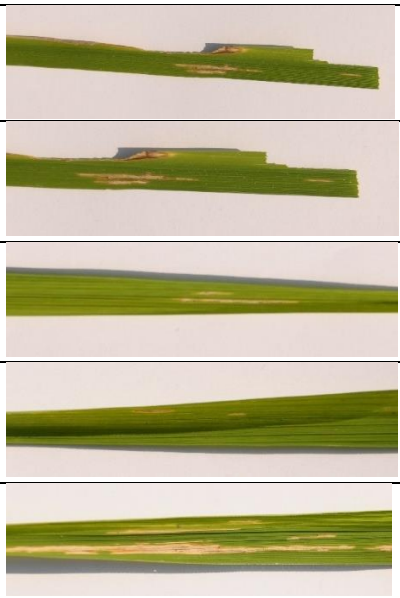
```
[ ] import splitfolders

[ ] # melakukan pembagian data training dan validasi
    splitfolders.ratio('/content/drive/MyDrive/Dataset_Skripsi/rice_leaf_diseases',
                       output="/content/drive/MyDrive/Dataset_Skripsi/split_result",
                       seed=1337, ratio=(.8, .2))

    Copying files: 120 files [00:28,  4.20 files/s]

[ ] # inisialisasi direktori data yang akan digunakan
    base_dir='/content/drive/MyDrive/Dataset_Skripsi/split_result'

[ ] # data training terletak pada path train didalam direktori utama
    train_dir = os.path.join(base_dir, 'train')
    # data validasi terletak pada path val didalam direktori utama
    val_dir = os.path.join(base_dir, 'val')

    # path data training apel Braeburn
    Bacterial_leaf_blight_dir_train = os.path.join(train_dir, 'Bacterial leaf blight')
    # path data training apel Brown spot
    Brown_spot_dir_train = os.path.join(train_dir, 'Brown spot')
    # path data training apel Leaf smut
    Leaf_smut_dir_train = os.path.join(train_dir, 'Leaf smut')
```

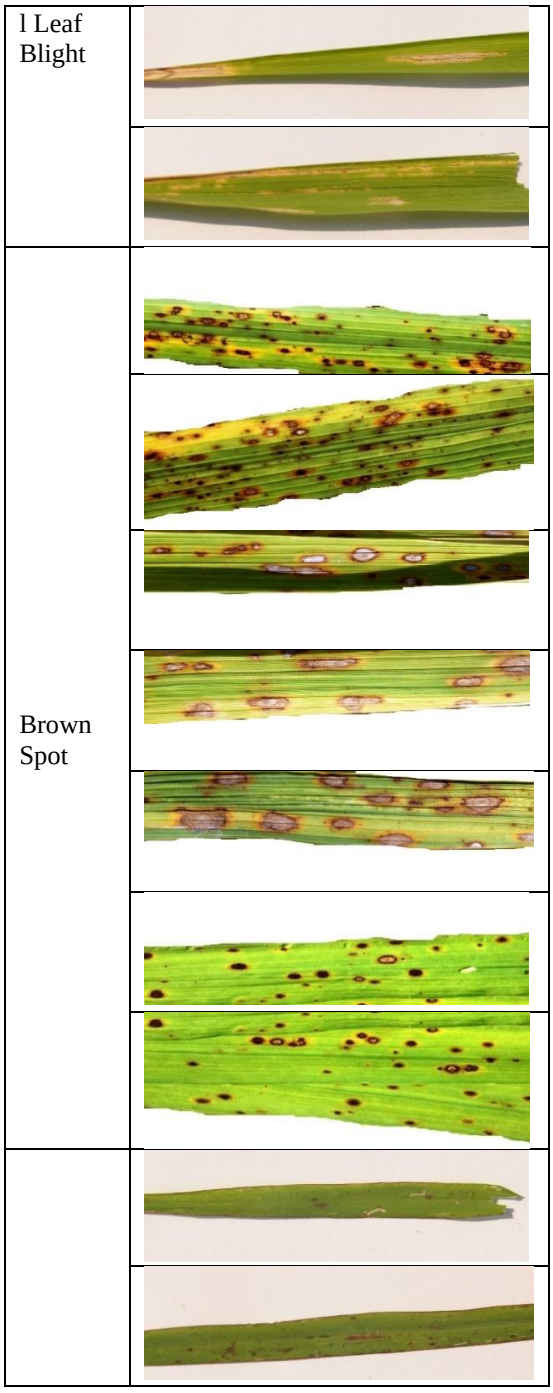
Gambar 16 Pembagian Direktori Data Training dan Validasi

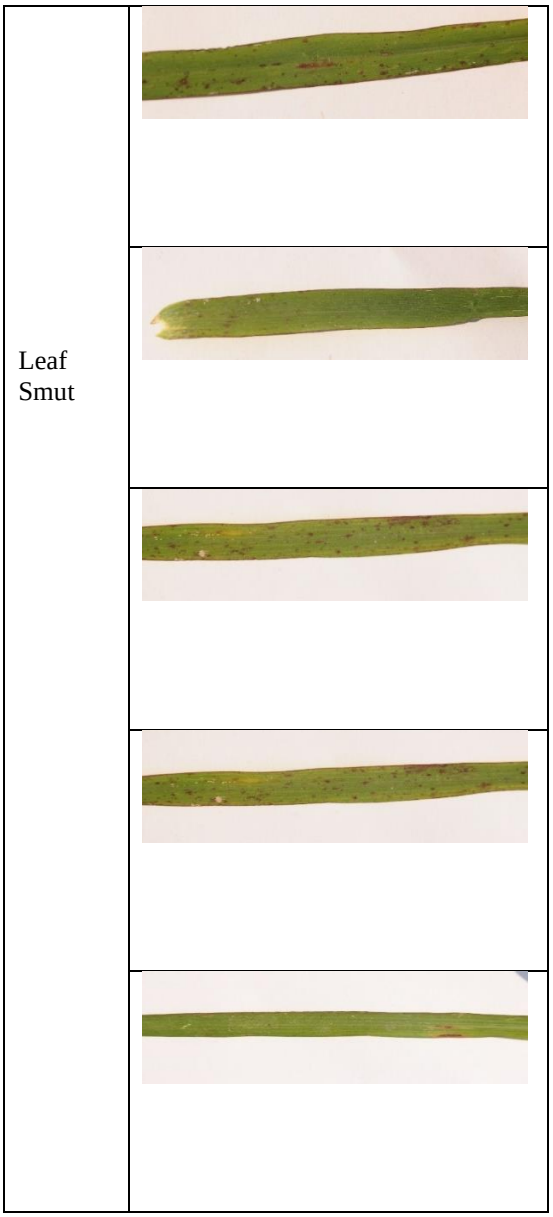
Implementasi Sistem

Pada penelitian ini dilakukan pengujian dengan menggunakan data citra digital sebanyak 120 data. 40 data jenis penyakit , 40 jenis penyakit dan 40 jenis penyakit yang sudah dilakukan *resize* menjadi ukuran 64 x 64 piksel. Berikut contoh gambar dataset training

Tabel 4 Jenis penyakit tanaman padi

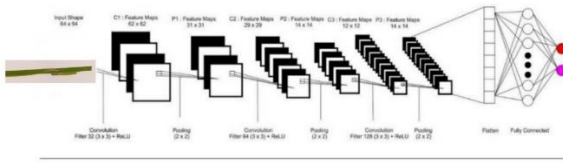
Jenis penyakit tanaman padi	Gambar
Bacteria	





Preprocessing dan Pembuatan Model CNN

Setelah melakukan sinkronisasi dan inisialisasi terhadap direktori penyimpanan dataset pada *google colab*, Langkah berikutnya yaitu tahap preprocessing dan pembuatan model CNN. Pada tahap ini, ukuran semua gambar pada input akan diubah menjadi 64 x 64 pixel. Kemudian dataset akan dilakukan training menggunakan algoritma CNN dengan model yang sudah dibuatkan sebagai berikut



Gambar 17 Arsitektur jaringan

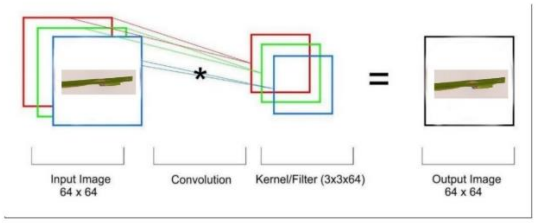
Gambar 17 merupakan arsitektur jaringan pada proses training untuk mendapatkan model yang optimal. Pada penelitian ini menggunakan input gambar dengan ukuran 64 x 64

1. Proses konvolusi pertama menggunakan kernel size 3 x 3 dan jumlah filter sebanyak 16 filter. Proses konvolusi adalah proses kombinasi dua buah matriks yang berbeda untuk menghasilkan suatu nilai matriks yang baru. Fungsi aktivasi *ReLU (Rectified Liniear Unit)* ditambahkan setelah proses *konvolusi* dengan bertujuan untuk mengubah nilai negative menjadi nol atau bisa juga dikatakan menghilangkan nilai negative dalam sebuah matriks hasil konvolusi. Hasil konvolusi pertama memiliki matriks baru berukuran 62 x 62
2. Proses *Pooling* pertama menggunakan kernel ukuran 2x2. Proses *pooling* merupakan proses pengurangan ukuran pada suatu matriks dengan menggunakan operasi *pooling.* Pooling layer terdiri dari filter dengan ukuran tertentu yang secara bergantian bergerser pada bagian *feature map*. Pada penelitian ini peneliti menggunakan *Max Pooling* pada proses *pooling*. Pada proses *Max Pooling* nilai maksimul dari matriks konvolusi pertama mengambil dari pergeseran kernel dengan stride yaitu 2. Berdasarkan hasil dari operasi *pooling*, dihasilkan matriks baru berukuran 31 x 31 dari hasil konvolusi pertama berukuran 62 x 62.
3. Proses konvolusi berikutnya dengan meneruskan hasil dari proses *pooling* sebelumnya yaitu dengan input matriks gambar sebesar 31 x 31 dengan jumlah filter sebanyak 64 dan ukuran kernel 3x3. Proses konvolusi kedua ini menggunakan fungsi dari aktivasi *ReLU* dan menghasilkan gambar berukuran 29x 29.
4. Setelah konvolusi selesai, proses berlanjut pada proses *pooling* kembali dimana proses ini hampir sama dengan proses *pooling* pertama. Hasil output didapatkan berdasarkan dari *poling* sebelumnya memiliki ukuran 14 x 14
5. Proses selanjutnya menuju konvolusi berikutnya, input matriks gambar berukuran 14 x 14 dengan jumla filter sebanyak 128 menggunakan kernel berukuran 3 x 3. Proses konvolusi kali ini juga menggunakan fungsi dari aktivasi *ReLU*. Hasil dari konvolusi ketiga ini adalah gambar dengan ukuran 12 x 12
6. Setelah konvolusi baru selesai lanjut menuju proses *pooling* berikutnya lagi. Pada proses ini ternyata hamper sama dengan pooling sebelumnya juga dan menghasilkan output berupa gambar berukuran 6 x 6
7. Selanjutnya adalah proses *Flatten* atau *Fully Connected*. Flatten disini untuk mengubah *output pooling* menjadi sebuah vector.
8. Menuju proses terakhir dengan menggunakan aktivasi fungsi *Softmax*.secara spesifik fungsi ini umumnya digunakan pada metode klasifikasi *multinominal logistic regression* dan *multiclass linear discriminant analysis*. Fungsi dari *sigmoid* sendiri akan mengubah nilai linear menjadi non-linear dan akan bernilai nol hingga satu.

Berdasarkan dari uraian arsitektur jaringan yang digunakan untuk proses pelatihan
Didapatkan total jumlah parameter pada model arsitektur sebanyak 361,635.

Pelatihan

Pada sub bab ini akan menjelaskan proses pelatihan pada data training dengan model CNN yang telah dibuat.



Gambar 18 Proses Konvolusi

A. Proses Convolution Layer

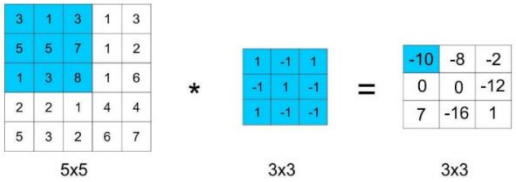
Berdasarkan penguraian dari arsiktektur

jaringan pada sub bab sebelumnya berikut ini adalah pembahasan mengenai proses konvolusi.

```
[ ]  model2 = tf.keras.models.Sequential([
        tf.keras.layers.Conv2D(16, (3,3), activation='relu',
                               input_shape=(64, 64, 3)),
        tf.keras.layers.MaxPooling2D(2, 2),
        tf.keras.layers.Conv2D(32, (3,3), activation='relu'),
        tf.keras.layers.MaxPooling2D(2, 2),
        tf.keras.layers.Conv2D(64, (3,3), activation='relu'),
        tf.keras.layers.MaxPooling2D(2,2),
        tf.keras.layers.Conv2D(128, (3,3), activation='relu'),
        tf.keras.layers.MaxPooling2D(2,2),
        tf.keras.layers.Flatten(),
        tf.keras.layers.Dense(512, activation='relu'),
        tf.keras.layers.Dense(3, activation='softmax')
    ])
```

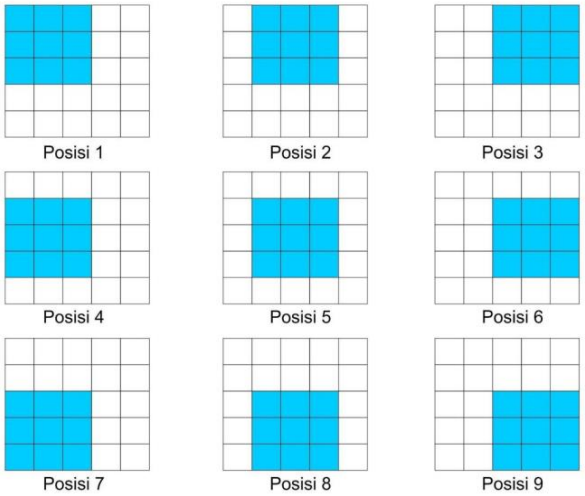
Gambar 19 Source Code Konvolusi

Konvolusi merupakan mengkombinasi dua buah deret angka yang menghasilkan deret angka yang ketiga. Jika diimplementasikan angka pada konvolusi ini adalah bentuk matrik *array.* Pada input, gambar memiliki ukuran piksel dari gambar sebesar 64 x 64 x 3, ini ,menunjukkan bahwa tinggi dan lebar piksel dari gambar sebesar 64 dan gambar tersebut memiliki 3 channel warna yaitu *red, green, dan blue* atau yang sering disebut RKGB. Pada setiap channel piksel pasti memiliki nilai matriks yang berbeda – beda. Input akan di konvo dengan nilai filter yang sudah ditentukan. Filter merupakan blok lain atau kubus dengan tinggi dan lebar yang lebih kecil namun kedalamannya sama dengan yang tersapu di atas gambar dasar atau gambar asli. Filter digunakan untuk menntukan pola apa yang akan dideteksi yang selanjutnya di konvolusi. atau dikaitkan dengan nilkai pada matriks input, nilai masing – masimg kolom dan baris pada matriks sangat bergantung pada jenis pola yang akan dideteksi. Jumlah filter pada konvo ini sebanyak 64 piksel dengan ukuran 3 x 3, ini berarti gambar yang dihasilkan dari konvolusi akan sebanyak 64 fitur map.
Supaya dapat lebih memahami cara kerja dari proses konvolusi, peneliti akan menggunakan sampel matriks pada *input image.* Karena pada penelitian ini menggunakan input ukuran 64 x 64 piksel maka peneliti hanya akan mengambil sebagai nilai matriks saja yang akan dijadikan sampel pada proses konvolusi.



Gambar 20 Perhitungan Proses Konvolusi

Gambar 19 menunjukkan proses konvolusi dengan menggunakan ukuran kernel 3 x 3, dengan menggunakan stride 1. Stride disini artinya jumlah pergeseran kernel terhadap matriks input berjumlah satu. Jika divisualisasikan sebagai berikut:
Gambar 20 menunjukkan perhitungan dot *product* pad proses konvolusi dimana sebuah kernel urkuran 3 x 3 yang dimulai pada sisi bagian kiri. Proses ini disebut dengan *sliding windows*, namun pada penelitian ini diberikan nilai *padding* 1, yaitu adanya penambahan nilai 0 di sekeliling nilai matriks input agar input dan output memiliki nilai matriks yang sama, sehingga tidak mengurangi informasi pada gambar. Proses ini dilakukan dari ujung kiri atas hingga sampai ujung kiri bawah.
Perhitungan dot product dapat dilihat sebagai berikut:



Gambar 21 Posisi Kernel pada Konvolusi

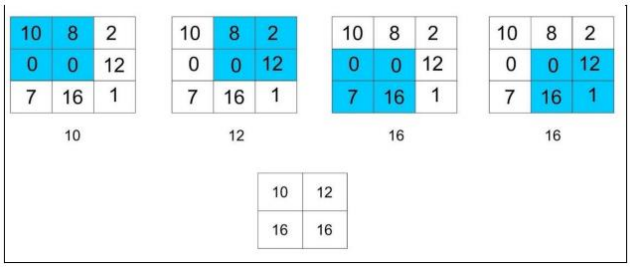
Posisi 1 = (3x1) + (5x(-1) + (1x1) + (1x(-1)) + (5x1) + (3x(- 1)) + (3x1) + (7x(-1)) + (8x1) = -10
Posisi 2 = (1x1) + (5x(-1)) + (3x1) + (3x (-1) + (7x1) + (8x(- 1)) + (1x1) + (1x(-1)) + (1x1) = -8
Posisi 3 = (3x1) + (7x(1)) + (8x1) + (1x(-1) + (1x1) + (1x(- 1)) + (3x1) + (2x(-1)) + (6x1) = -2

Posisi 4 = (5x1) + (1x(-1)) + (2x1) + (5x(-1)) + (3x1) + (2x(- 1)) + (7x1) + (8x(-1)) + (1x1) = 0
. Posisi 5 = (5x1) + (3x(1)) + (2x1) + (7x(-1)) + (8x1) + (1x(- 1)) + (1x1) + (1x(-1)) + (4x1) = 0
Posisi 6 = (7x1) + (8x(1)) + (1x1) + (1x(-1) + (1x1) + (4x(- 1)) + (2x1) + (6x-1) + (4x1) = -12
Posisi 7 = (1x1) + (2x(-1)) + (5x1) + (3x(-1)) + (2x1) + (3x(- 1)) + (8x1) + (1x(-1)) + (2x1) = 7
Posisi 7 = (1x1) + (2x(-1)) + (5x1) + (3x(-1)) + (2x1) + (3x(- 1)) + (8x1) + (1x(-1)) + (2x1) = 7
Posisi 9 = (8x1) + (1x(-1) + (2x1) + (1x(-1)) + (4x1) + (6x(- 1)) + (6x1) + (4x(-1)) + (7x1) = 1

Kemudian sebelum dilanjutksan ke proses pooling layer, untuk menghilangkan nilai negative pada hasil, pada arsitektur jangan digunakan aktivasi ReLU (*Rectifed Linier Unit*) setelah proses konvolusi. Fungsi dari aktivasi tersebut adalah melakukan "*threshold*" dari 0 hingga *infinity*. Nilai yang ada pada hasil konvolusi yang bernilai n egatif akan diubah dengan aktivasi tersebut menjadi nol dan nilai yang lainnya sampai *infinity*.

B. Proses *Pooling*

Pooling merupakan pengurangan ukuran matriks dengan menggunakan operasi *pooling* (penggabungan). Metode yang digunakan dalam proses pooling ini max *pooling*. Dalam penelitian [12] menunjukkan bahwa penggunan metode *max-pooling* lebih bunggul disbanding dengan metode ini menjadi salah satu metode terbaik dalam proses pooling. Berikut ini gambar dari proses *pooling* :

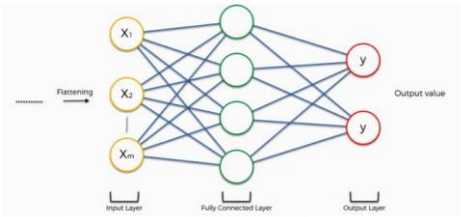


Gambar 22 Proses Pooling

Proses pooling ini menggunakan ukuran 2 x 2 dengan stride 1 dimana jumlah pergeseran karnel terhadap matriks input berjumlah satu. Dalam proses pooling ini digunakan metode *max-pooling*, dimana window akan bergeser sesuai dengan ukurannya dan juga stridenya agar mendapatkkan nilai yang paling maksimum. Dapat dilihat pada gambar x.x output dari proses ini memiliki nilai paling maksimum yang diambil dari matriks fitur map hasil konvolusi, kemudian hasil dari *max-pooling* tersebut berukuran 2 x 2.

C. Proses Fully Connected.

Proses selanjutnya adalah Fully Connected layer. Ini bertujuan untuk melakukan transformasi pada dimensi data agar dapat diklasifikasikan secara linear

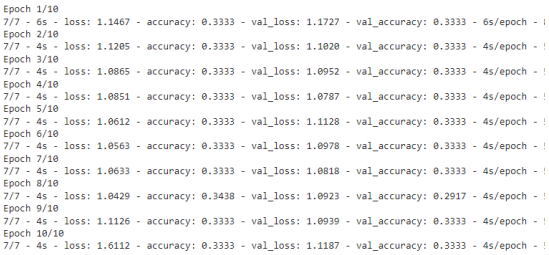


Gambar 23 Proses Fully Connected Layer

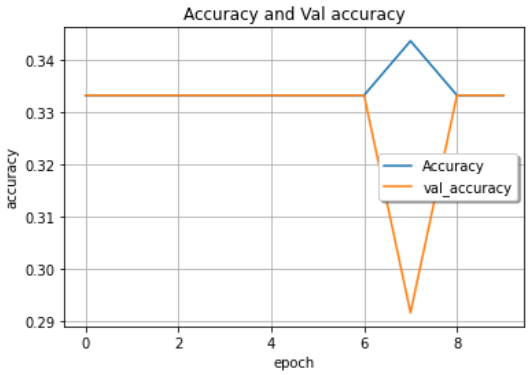
Gambar 23 merupakan proses converting hasil dari fitur map maxpooling menjadi flatten atau vektor. Dalam proses ini nilai input matriks dari layer sebelumnya akan diubah menjadi vektor. Proses ini sama dengan proses MLP (Multilayer Perceptron). Jaringan ini pada umumnya menggunakan lapisan yang terhubung sepenuhnya di mana setiap piksel dianggap sebagai neuron terpisah. Dalam proses ini biasanya diterapkan metode "dropout". Metode ini bertujuan untuk menonaktifkan beberapa edge yang terhubung ke setiap neuron untuk menghindari overfitting. Setelah itu proses rakhir adalah klasifikasi. Dalam proses ini gunakan aktivasi fungsi softmax. Aktivasi ini an membantu MLP untuk mengklasifikasi put terhadap targetnya, yaitu kedalam 3 kelas nis penyakit tanaman padi (Bacterial leaf ight, brown spot dan leafsmut).

D. Hasil Training

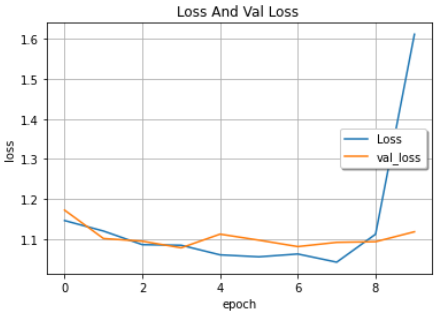
etelah melewati beberapa proses dalam goritma Convolutional Neural Network (CNN) didapatkan hasil training dan validation. Proses ini menggunakan jumlah 20 epoch, nilai learning rate 0.001, berikut grafik hasil training dan validation menggunakan matplotlib :



Gambar 24 Hasil akurasi



Gambar 25 Grafik Accuracy dan Val accuracy



Gambar 26 Grafik loss dan val loss

Berdasarkan gambar 26 Proses training disini menggunakan learning rate 0.001 dengan input gambar sebesar 64 x 64 piksel. Waktu pelatihan yang dibutuhkan untuk 10 epoch dalam menjalankan training model yaitu 42 detik. Semakin lama epoch maka semakin lama juga waktu yang akan dibutuhkan untuk melakukan proses *training* model. Kemudian *accuracy* dari data validationyang didapatkan mencapai dengan nilai 33% loss sebesar 1.1006

E. Hasil Testing

Pada proses testing menggunakan data uji sebanyak 120 data. Untuk setiap kelas jenis penyakit tanaman padi sebanyak 40 gambar. Hasil confusion matriks adalah sebagai berikut.

Tabel 5 Confusion Matriks

Matriks		Predict class		
		Bacterial leaf - blight	Brown spot	Leaf smut
Actual class	Bacterial leaf blight	38	2	0
	Brown Spot	17	23	0
	Leaf smut	16	20	4

Berdasarkan tabel diatas hasil prediksi dari model terhadap data testing data baru menunjukan hasil kurang baik. Prediksi terhadap jenis penyakit Bacterial leaf bligth diklasifikasikan ke dalam Bacterial leaf bligth, ini artinya klasifikasi terhadap gambar tersebut adalah benar. Prediksi pada jenis buah apel yang kedua pink lady diklasifikasikan benar sebagai bacterial leaf blight sebanyak 38 dan missing data dari input bacterial leaf blight diklasifikasikan sebagai Brown spot sebanyak 2 data dan leaf smut sebanyak 0. Perhitungan menggunakan metode overall Accuracy keseluruhan matriks diatas adalah sebagai berikut:

Overall accuracy =65/120 = 54,1%

Jadi akurasi yang dihasilkan oleh model dengan input gambar 64 x 64 piksel dengan nilai learning rate sebesar 0,001 dan jumlah sampel testing 120 data didapatkan nilai akurasi sebesar 54,1%

Penentuan Parameter

Penentuan model terbaik, harus dicari nilai terbaik parameter dalam model CNN, parameter yang dimkasud adalah pengaruh jumlsh *epoch.* Pengaruh ukuran ukuran input gambar, pengaruh jumlah data training,

pengharuh jumlah scenario data, ukuran kernel dan *learning rate.*tujuan dari penentuan parameter model ini adalah ingin membadningkan model mana yang paling terbaik dengan memperhatikan nilai parameternya.

Pengaruh nilai *Learning Rrate*

Learning rate pada penelitian ini juga melakukan uji coba dengan menggunakan nilai *Learning Rate* yang berbeda dan epoch 10. Dalam klasifikasi gambar pada umumnya banyak sekali penggunaan nilai *learning rate* sebesara 0.1 sampai 0.0001[7]. Penentuan *learning rate* biasanya ditentukan oleh peneliti sendiri dan tidak terpaku oleh nilai tertentu. Pada penelitian ini peneliti menggunakan tiga nilai yaitu 0.1, 0.01, 0.001. penentuan nilai dari *learning rate* ini berpengaruh sekali pada peforma akurasi. Berikut ini adalah Hasil perbandingan hasil training dan epoch.

Tabel 6 Akurasi berdasarkan *Learning rate*

Learning rate	*Val Accuracy*	*Loss Validation*	*Time (seconds)*
0,1	33,3%	1.1006	41
0,01	33,3%	1.0987	43
0,001	33,3%	1.1001	41

Berdasarkan tabel 4.4 penggunaan learning rate 0.1 menghasilkan tingkat akurasi yang tidak optimal yaitu sebesar sekian dan nilai *loss* sebesar sekian, lalu Ketika penggunaan *learning rate* sebesar 0.01 hasilnya sekian. Ini dikarenakan ketikan penggunaan *learning rate* menggunakan nilai cukup besar, maka nilai *loss* akan semakin meningkat Ketika menjalankan beberapa iterasi pada saat dilakukan *training*, setelah itu pada nilai *learning rate* 0.001 yang digunakan pada penelitian ini menghasilkan nilai akurasi sebesar sekian dan nilai loss sebesar sekian. Terjadi naik turun Ketika penggunaan masing-masing learning rate yang sama hanya pada *val accuracy*.

Pengaruh nilai *Epoch*

Epoch adalah seluruh dataset sudah melewati proses training pada *Neural Network* sampai dikembalikan ke awal dalam satu putaran. Pada *Neural Network* satu epoch itu terlalu besar dalam proses pelatihan karena terlalu besar dalam proses pelatihan karean semua data masuk kedalam proses *training* sehingga akan membuthkan waktu cukup lama. Biasanya untuk mempermudah proses training, dataset dibagi menjadi satuan atau per *Batch* (*Batch Size)*.

Tabel 7 Akurasi berdasarkan *Epoch*

Epoch	*Val Accuracy*	*Loss Validation*	*Time (seconds)*
10	41,6%	100%	36
20	66,6%	100%	29
30	70,8%	96,8%	53

Berdasarkan tabel diatas dengan menggunakan nilai learning rate 0.001 didapatkan akurasi yang sama pada setiap nilai epoch yang berbeda. Jika dilihat dari tabel semakin tinggi nilai epoch yang digunakan maka nilai loss akan semakin kecil.

SIMPULAN

Model CNN pada penelitian ini menggunakan input shape dengan ukuran 64 x 64 memiliki nilai learning rate 0.001, ukuran filter 3 x 3, jumlah epoch 20 data training sebanyak 140 , menghasilkan tingkat akurasi training dan validation dalam melakukan klasifikasi gambar citra jenis penyakit tanaman padi sebesar 33% pada training dan 70,8% pada validation.

Penelitian ini menggunakan data testing baru untuk diujikan pada model yang sudah dibuat. Hasil testing didapatkan tingkat akurasi baru dalam melakukan klasifikasi gambar citra jenis penyakit tanaman padi sebesar 54,1% antara bacterial leaf blight, brown spot, dan leaf smut.

Semakin besar nilai epoch semakin lama proses training berjalan tetapi tidak begitu mempengaruhi akurasi

Learning rate tidak mempengaruhi tingkat akurasi pada saat training

dan menggunakan arsitektur model lain.

DAFTAR PUSTAKA

[1] ー.“Luas Panen dan Produksi Padi di Indonesia 2020 (Angka Sementara)” 15 Oktober 2020. [Online]. Tersedia pada: https://www.bps.go.id/pressrelease/download.html?nrbvfeve=MTc1Nw%3D%3D&sdfs=ldjfdifsdjkfahi&twoadfnoarfeauf=MjAyMC0xMS0wMyAwMDozNDowNg%3D%3D. [Diakses 30 Oktober 2020].

[2] W. Febrianto, A. H. Suryatama, N. Afrianto, I. Mualana, P. N. Hidayat. “Analisis Sistem Pakar untuk Mengidentifikasi Penyakit dan Hama pada Tanaman Padi dengan Metode Bayes”. Universitas Amikom Yogyakarta, 2018.

[3] Tri Audia Lestari. “Pengamatan Penyakit-Penyakit Tanaman Padi di Desa Sako Kecamatan Rambutan Kabupaten Banyuasin Dengan Sistem Jajar Legowo” [Skripsi]. Fakultas Pertanian Universitas Sriwijaya. 2019.

[4] S. Ghosal dan K. Sarkar. “Rice Leaf Diseases Classification Using CNN With Transfer Learning”. IEEE Calcutta Conference (CALCON), 2020.

[5] A. K. Singh, Rubiya .A, B. S. Raja. “Classification of Rice Disease Using Digital Image Processing and SVM Classifier”. International Journal of Electrical and Electronics Engineers, Vol. 7,2015.

[6] F. F. Maulana and N. Rochmawati, “Klasifikasi Citra Buah Menggunakan Convolutional Neural Network,” J. Informatics Comput. Sci., vol. 01, pp. 104–108, 2019, [Online]. Available: jurnalmahasiswa.unesa.ac.id › article.

[7] N. Triaono "IMPLEMENTASI DEEP LEARNING UNTUK IMAGE CLASSIFICATION MENGGUNAKAN ALGORITMA CONVOLUTIONAL NEURAL NETWORK (CNN) PADA CITRA WAYANG GOLEK" DSpace UII Statistic [319], 2018

[8] M. A. Pangestu and H. Bunyamin, “Analisis Performa dan Pengembangan Sistem Deteksi Ras Anjing pada Gambar dengan Menggunakan PreTrained CNN Model,” Tek. Inform. dan Sist. Inf., vol. 4, no. 2, pp. 337– 344, 2018.

[9] A. Y. Wicaksono, N. Suciati, C. Fatichah, K. Uchimura, and G. Koutaki, “Modified Convolutional Neural Network Architecture for Batik Motif Image Classification,” IPTEK J. Sci., vol. 2, no. 2, pp. 26–30, 2017, doi: 10.12962/j23378530.v2i2.a2846.

[10] IdWeb S. E. Limantoro, Y. Kristian, and D. D. Purwanto, “Deteksi Pengendara Sepeda Motor Menggunakan Deep Convolutional Neural Networks,” Semin. Nas. Teknol. Inf. dan Komun., pp. 79–86, 2017.

[11] P. N. Rena "Penerapan Metode Convolutional Neural Network pada Pendeteksi Gambar Notasi Balok" Fakultas Sains dan Teknologi 61 Universitas Islam Negeri Syarif Hidayatullah Jakarta xiv, 66 hlm; 28 cm, 2019

[12] M. Zufar and B. Setiyono, “Convolutional Neural Networks Untuk Pengenalan Wajah Secara Real-time,” J. Sains dan Seni ITS, vol. 5, no. 2, p. 128862, 2016.

[13] W. S. Eka Putra, “Klasifikasi Citra Menggunakan Convolutional Neural Network (CNN) pada Caltech 101,” J. Tek. ITS, vol. 5, no. 1, 2016, doi: 10.12962/j23373539.v5i1.15696.