KLASIFIKASI PENYAKIT PADA DAUN KENTANG MENGGUNAKAN PENGOLAHAN CITRA DENGAN METODE CONVOLUTIONAL NEURAL NETWORK (CNN)

TUGAS AKHIR

Tugas Akhir ini sebagai salah satu syarat untuk memperoleh gelar sarjana Informatika

Universitas Pembangunan Nasional "Veteran" Yogyakarta

Disusun Oleh :

Torangto Yos P. Situngkir

123180122

PROGRAM STUDI INFORMATIKA

JURUSAN INFORMATIKA

FAKULTAS TEKNIK INDUSTRI

UNIVERSITAS PEMBANGUNAN NASIONAL "VETERAN"

YOGYAKARTA

2022

HALAMAN PENGESAHAN PEMBIMBING

KLASIFIKASI PENYAKIT PADA DAUN KENTANG MENGGUNAKAN PENGOLAHAN CITRA DENGAN METODE CONVOLUTIONAL NEURAL NETWORK (CNN)

Disusun oleh:

Torangto Yos P. Situngkir
123180122

Telah diuji dan dinyakatan lulus oleh pembimbing
pada tanggal : 4 Oktober 2022

Menyetujui,

Pembimbing I	Pembimbing II
Bambang Yuwono, S.T., M.T.	Yuli Fauziah, S.T., M.T.
NIDN. 0512027401	NIDN. 0508077102

Mengetahui,
Koordinator Program Studi

Dr. Heriyanto, A.Md, S.Kom., M.Cs.
NIDN. 0508067703

HALAMAN PENGESAHAN PENGUJI

KLASIFIKASI PENYAKIT PADA DAUN KENTANG MENGGUNAKAN PENGOLAHAN CITRA DENGAN METODE CONVOLUTIONAL NEURAL NETWORK (CNN)

Disusun oleh:

Torangto Yos P. Situngkir
123180122

Telah diuji dan dinyatakan lulus oleh penguji
pada tanggal : 4 oktober 2022

Menyetujui,

Penguji I	Penguji II
Bambang Yuwono,S.T.,M.T	Yuli Fauziah, S.T., M.T
NIDN. 0512027401	NIDN. 0508077102
Penguji III	**Penguji IV**
Budi Santosa,S.Si.,M.T	Herry Sofyan,S.T.,M.Kom.
NIDN. 0510097001	NIDN. 0524046402

SURAT PERNYATAAN KARYA ASLI TUGAS AKHIR

Sebagai mahasiswa Program Studi Informatika Fakultas Teknik Industri Universitas Pembangunan Nasional "Veteran" Yogyakarta, yang bertanda tangan di bawah ini, saya:

Nama : Torangto Yos P. Situngkir
NIM : 123180122

Menyatakan bahwa karya ilmiah saya yang berjudul :

KLASIFIKASI PENYAKIT PADA DAUN KENTANG MENGGUNAKAN PENGOLAHAN CITRA DENGAN METODE CONVOLUTIONAL NEURAL NETWORK (CNN)

Merupakan karya asli saya dan belum pernah dipublikasikan dimanapun. Apabila ada di kemudian hari, karya saya disinyalir bukan merupakan karya asli saya, maka saya bersedia menerima konsekuensi apa pun yang diberikan Program Studi Informatika Fakultas Teknik Industri Universitas Pembangunan Nasional "Veteran" Yogyakarta kepada saya.

Demikian surat pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Yogyakarta
Pada tanggal :
Yang menyatakan,

(Torangto Yos P. Situngkir)

PERNYATAAN BEBAS PLAGIAT

Saya yang bertanda tangan di bawah ini :

Nama : Torangto Yos P. Situngkir
NIM : 123180122
Fakultas : Teknik Industri/Informatika

Dengan ini menyatakan bahwa judul Tugas Akhir :

KLASIFIKASI PENYAKIT PADA DAUN KENTANG MENGGUNAKAN PENGOLAHAN CITRA DENGAN METODE CONVOLUTIONAL NEURAL NETWORK (CNN)

Adalah hasil kerja saya sendiri dan benar bebas dari plagiat kecuali cuplikan serta ringkasan yang terdapat di dalamnya setalah saya jelaskan sumbernya (sitasi) dengan jelas. Apabila pernyataan ini terbukti tidak benar maka saya bersedia menerima sanksi sesuai peraturan Mendiknas RI No. 17 Tahun 2000 dan Pernyataan dan Perundang-undangan yang berlaku.

Demikian surat pernyataan ini saya buat dengan penuh tanggung jawab.

Dibuat di : Yogyakarta
Pada tanggal :
Yang menyatakan,

(Torangto Yos P. Situngkir)

ABSTRAK

Kentang (*Solanum tuberosum*) merupakan salah satu tanaman pangan terpenting ketiga di dunia, setelah gandum dan beras.Dalam mengelola pertanian kentang tentu saja ada beberapa kendala diantaranya adalah penyakit yang menyerang pada daun kentang yang jika dibiarkan dapat menyebabkan berkurangnya hasil panen atau bahkan gagal panen.Penyakit yang paling sering menyerang tanaman kentang ini adalah penyakit pembusukan dini (*early blight*) dan penyakit busuk daun (*late blight*).Kedua penyakit ini terkadang terlihat mirip sehingga sulit untuk dikenali. Dengan perkembangan teknologi saat ini, memungkinkan dilakukannya deteksi penyakit terhadap tanaman secara otomatis menggunakan komputer.Salah satunya menggunakan *Deep Learning,* Metode *Deep Learning* yang saat ini memiliki hasil paling signifikan dalam pengenalan citra adalah *Convolutional Neural Network* (CNN).

Penelitian ini menggunakan metode CNN dengan model paling optimal.Sebelum dilakukan klasifikasi terlebih dahulu dilakukan proses *preprocessing* dengan menghapus *background* pada citra daun kentang.Selanjutnya dilakukan pengujian kombinasi *Hyperparameter* dan *Optimizer. Hyperparameter* yang diujikan antara lain *epoch* dan *convolutional layer* kemudian algoritma optimasi yang diujikan yaitu Adam,SGD,dan RMSProp.Pengujian dilakukan untuk menemukan model dengan tingkat akurasi tertinggi dan nilai *loss* paling rendah,penelitian ini melakukan pengujian pada 18 jenis model yang telah dirancang.Setelah itu dilakukan pengembangan sistem dengan pengujian citra gambar yang sudah di augmentasi menggunakan rotasi dengan besar 30° dan 60°,*brightness* ditambah 40% dan dikurangi 40%,*shear range* 15% dan 30%.

Pengujian dilakukan dengan skenario *confusion matrix* untuk menentukan tingkat akurasi yang dihasilkan oleh model CNN. Berdasarkan dari hasil klasifikasi yang dilakukan,didapatkan tingkat akurasi sebesar 98,2% dengan model paling optimal adalah *Optimizer* Adam, *Convolutional Layer* 5,dan *epochs* 100.

Kata kunci : Daun Kentang,*Convolutional Neural Network*,Klasifikasi,*Hyperparameter*

ABSTRACT

Potato (*Solanum Tuberosum*) is the third most important food crop in the world, after cereals and rice. In managing this potato, of course, there are several obstacles including diseases that attack potato leaves which if left unchecked will result in poor production or even crop failure.These potatoes are early blight and late blight. Both of these diseases may look like diseases that are difficult to recognize. With current technological developments, it is possible to detect diseases of plants automatically using a computer. One of them uses Deep Learning, the Deep Learning Method which currently has the most significant results in image recognition is the Convolutional Neural Network (CNN).

In this study using the CNN method with the most optimal model. Before the classification is carried out, the preprocessing process is carried out by removing the background on the potato leaf image. Then, the combination of Hyperparameter and Optimizer was tested. The tested hyperparameters include the epoch and convolutional layer then the optimization algorithms tested are Adam, SGD, and RMSProp. Tests were carried out to find the model with the highest level of accuracy and the lowest loss value, this study tested 18 combinations of models that have been designed. After that, the system was developed by testing the augmented images using a rotation with a size of 30° and 60°, brightness plus 40% and reduced 40%, shear range 15% and 30%.

The test is carried out with a confusion matrix scenario to determine the level of accuracy generated by the CNN model. Based on the results of the classification carried out, an accuracy rate of 98.2% was obtained with the most optimal model being the Adam Optimizer, Convolutional Layer 5, and 100 epochs.

Keywords : Potato Leaves,Convolutional Neural Network,Classification,Hyperparameter

KATA PENGANTAR

Segala puji syukur saya ucapkan kepada Tuhan Yang Maha ESA atas berkat rahmat dan karunia-Nya saya dapat menyelesaikan Tugas Akhir ini dengan baik.Tugas akhir ini sebagai salah satu syarat untuk menyelesaikan program sarjana (S1) di Program Studi Informatika Fakultas Teknik Industri Universitas Pembangunan Nasional "Veteran" Yogyakarta.

Dalam penulisan tugas akhir ini saya menyadari banyak pihak yang telah memberikan sumbangan baik pikiran,waktu,tenaga,bimbingan dan dorongan kepada saya sehingga akhirnya tugas akhir ini dapat selesai.Oleh karena itu, saya persembahkan karya ini berserta ucapan terimakasih kepada :

1. Tuhan Yang Maha Esa atas penyertaan dari awal hingga akhir masa perkuliahan sehingga saya dapat menyelesaikan Tugas Akhir dengan baik.
2. Papa, Mama, dan Adek saya yang selalu memberikan semangat dan kasih sayang mereka
3. Bambang Yuwono, S.T., M.T., selaku dosen pembimbing I yang telah memberikan bimbingan,arahan dan petunjuk selama penulisan tugas akhir
4. Yuli Fauziah, S.T., M.T., selaku dosen pembimbing II yang telah memberikan bimbingan,arahan dan petunjuk selama penulisan tugas akhir
5. Sahabat-sahabat saya selama masa perkuliahan Khalil, Rifqi, Edlan, Julian, Agave, Ridwan, dan Can Simbolon yang selalu memberikan support kepada saya baik dalam suka maupun duka, serta telah menemani saya dalam menyelesaikan tugas akhir ini, semoga persahabatan ini tidak pernah terputus sampai kapanpun
6. Sahabat-sahabat SMA yang sudah saya anggap seperti saudara/i Axel, Erikson, Gunawan, Evi, Hana, yang selalu memberikan semangat serta selalu ada dimasa sulit maupun bahagia, serta telah menemani saya dalam menyelesaikan tugas akhir ini.Semoga persahabatan ini tidak pernah terputus sampai kapanpun
7. Teman-teman angkatan 2018 Program Studi Informatika yang menjadi tempat bertukar ilmu dan bertukar informasi terkait akademik maupun non akademik
8. Serta semua pihak-pihak lain yang tidak dapat saya sebutkan satu persatu yang telah banyak mendoakan ,memberi motivasi serta membantu saya dalam penyusunan tugas akhir ini.

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

DAFTAR ALGORITMA

BAB I PENDAHULUAN

1.1 Latar Belakang

Kentang merupakan salah satu tanaman pangan terpenting ketiga di dunia, setelah gandum dan beras. Produksi global melebihi 300 juta metrik ton dan merupakan penyedia nutrisi dan kalori penting bagi umat manusia (Oppenheim & Shani, 2017). Di Indonesia, kentang dapat dimanfaatkan sebagai sayuran atau diolah menjadi bahan baku industri, seperti keripik kentang. Tentu saja ada beberapa kendala dalam mengelola pertanian kentang ini. Contohnya adalah penyakit yang menyerang daun kentang yang jika dibiarkan dapat menyebabkan berkurangnya hasil panen atau bahkan gagal panen (Rozaqi et al., 2021). Produksi kentang terancam oleh beberapa penyakit yang mengakibatkan kehilangan hasil yang cukup besar, serta menyebabkan penurunan kualitas serta kenaikan harga kentang (Taylor et al., 2008). Deteksi dini penyakit pada tanaman sangat penting bagi petani untuk mengendalikan penyebaran penyakit (Martinelli et al., 2015). Pemberian Tindakan sangat tergantung pada identifikasi penyakit yang menyerang tanaman.

Dalam ilmu biologi, daun digunakan sebagai indikator kesehatan tanaman dengan mengamati warna daun secara visual, yang berhubungan langsung dengan kandungan klorofil yang dikandungnya (Ali et al., 2019a). Karena sebagian besar gejala penyakit dapat dilihat pada daun, maka pengenalan gejala awal pada daun diharapkan dapat membantu proses pengendalian penyakit pada tanaman (Permadi & Harjoko,2015).Jamur dan bakteri merupakan penyebab penyakit pada daun kentang.Penyakit busuk daun (*late blight*) dan penyakit pembusukan dini (*early blight*) adalah penyakit yang disebabkan jamur. Penyakit busuk daun dapat menyebar ke bagian lain dari tanaman kentang, seperti tangkai, batang dan umbi kentang. Sehingga, petani harus memangkas daun yang terkena penyakit busuk daun lebih awal untuk mendapatkan panen yang baik (Rakhmawati dkk, 2018). Tentu saja penyakit ini dapat menimbulkan kerugian yang besar karena menurunkan kualitas dan kuantitas umbi kentang. Untuk mengantisipasi serangan lebih lanjut dan melakukan tindakan pengendalian, perlu diketahui jenis penyakitnya.

Oleh sebab itu, dalam penelitian ini dibangun sebuah sistem yang dapat mengklasifikasi citra daun kentang untuk memudahkan deteksi dini penyakit kentang khususnya bagi petani dan industri pertanian agar penyakit tersebut tidak menyebar dan berkembang lebih buruk. Perkembangan ilmu pengetahuan saat ini telah mendorong ditemukannya cara untuk mendeteksi penyakit tanaman secara otomatis dengan bantuan komputer. Deteksi yang dihasilkan komputer diyakini cukup akurat, sehingga deteksi penyakit dengan bantuan komputer cukup direkomendasikan (Tsany, A., & Dzaky, R. 2021). Salah satunya menggunakan Jaringan Syaraf Tiruan (JST) yang terinspirasi dari jaringan syaraf manusia. Konsep ini kemudian dikembangkan lebih lanjut dalam *Deep Learning*. Saat ini, *Deep Learning* telah menjadi salah satu topik hangat di dunia *Machine Learning* karena kapabilitasnya yang signifikan untuk memodelkan data yang kompleks seperti citra dan suara. Metode *deep learning* yang saat ini memiliki hasil paling signifikan dalam pengenalan citra adalah *Convolutional Neural Network* (CNN), karena CNN diimplementasikan berdasarkan sistem pengenalan citra pada visual cortex manusia,yaitu bagian pada otak yang bertugas untuk memroses informasi dalam bentuk visual (Suartika et al. 2016).

Beragam penelitian telah dilakukan dalam mengklasifikasikan penyakit daun dengan menggunakan teknik pengolahan citra.Teknik *Fuzzy K-Nearest neighbour* (FKNN) dalam mendeteksi penyakit tanaman pada rambutan (Rahayu & Mendes, 2021) dengan akurasi yang diperoleh 67%.Penelitian lain oleh (Sari, 2016) deteksi penyakit pada jagung dengan metode *Color Moments dan* GLCM diperoleh akurasi 89,375 %.Selanjutnya menggunakan *Convolutional Neural Network* (CNN) untuk deteksi penyakit pada daun cabai (Tsany & Dzaky, 2021) dengan akurasi yang dihasilkan mencapai 90%.Penelitian lainnya klasifikasi penyakit pada daun kentang menggunakan *Support Vector Machine* (SVM) berdasarkan fitur tekstur dan fitur warna oleh (Rakhmawati et al., 2018a).Namun belum banyak penelitian tentang klasifikasi penyakit pada daun kentang menggunakan *Convolutional Neural Network* (CNN).Penelitian pernah dilakukan oleh (Rozaqi et al., 2021) dengan akurasi yang diperoleh 94%. Dari penelitian yang sudah dilakukan sebelumnya dapat dikatakan bahwa CNN menghasilkan tingkat akurasi paling signifikan dalam pengenalan citra digital.

Oleh karena itu penelitian ini menggunakan metode CNN untuk mengklasifikasikan citra daun kentang berdasarkan jenis penyakitnya.Penelitian ini dilakukan untuk meningkatkan hasil dari penelitian khususnya oleh Rozaqi. Penelitian Rozaqi belum menggunakan metode untuk menentukan kombinasi parameter terbaik untuk mendapatkan model CNN.Maka dari itu,penelitian ini dilakukan untuk mengusulkan model CNN dengan mencari kombinasi *hyperparameter* dan algoritma optimasi terbaik. *Hyperparameter* yang diujikan adalah nilai *Epoch*,dan *Convolutional Layer* pada model yang diusulkan,karena kinerja prediktif sangat dipengaruhi oleh nilai *hyperparameter* ataupun *Optimizer* yang digunakan untuk melatihnya,penelitian ini telah dilakukan oleh (Wahyu Nugraha, 2022).Penelitian ini juga menggunakan *preprocessing* data dengan memisahkan objek daun dengan latar belakang yang dilakukan dengan bantuan *software image preprocessing*. Penyakit daun kentang yang akan diklasifikasikan ada 3 kelas, yaitu *Early Blight*, *Late Blight* dan *Healthy*.

1.2 Rumusan Masalah

Permasalahan yang di angkat dalam penelitian ini adalah sebagai berikut:

1. Belum adanya klasifikasi penyakit pada daun kentang menggunakan metode CNN dengan menguji kombinasi *hyperparameter* dan *optimizer* yang diberikan pada model CNN untuk menentukan kombinasi paling optimal
2. Mendeteksi penyakit pada daun kentang dan mengukur tingkat akurasi yang diperoleh dari proses pengklasifikasian citra penyakit daun kentang dengan menggunakan metode CNN

1.3 Batasan Masalah

Agar pembahasan masalah menjadi lebih jelas dan terarah, maka diperlukan adanya batasan masalah. Adapun ruang lingkup dalam susunan ini dibatas pada :

1. Klasifikasi terbatas pada penyakit daun kentang yaitu Pembusukan Dini(*Early Blight*),Busuk Daun(*Late Blight*) dan Sehat (*Healthy*)

2. Data yang digunakan dalam penelitian ini adalah dataset *Plant Village* yang berisi citra daun penyakit tanaman kentang dari kaggle.
3. Citra yang digunakan adalah RGB dan memiliki ukuran 256x256 pixel.
4. *Preprocessing* untuk menghilangkan latar belakang pada citra daun dengan aplikasi *background remover*
5. Hasil pengelompokan hanya berdasarkan sistem klasifikasi tanpa bantuan ahli penyakit tanaman.
6. Model dari *deep learning* dibuat dengan menggunakan bahasa pemrograman Python dengan menggunakan Google Colab dengan library Numpy,Pandas,Tensorflow,dan Keras.
7. *Hyperparameter* yang diujikan adalah *epoch,convolutional layer* dan pemilihan *optimizer* (Adam,SGD,&RMSProp)

1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah sebagai berikut :

1. Mengetahui model CNN yang mampu mengklasifikasikan jenis penyakit berdasarkan citra daun kentang dengan kombinasi *Hyperparameter* dan *Optimizer*.
2. Mengetahui hasil akurasi yang diperoleh dari proses klasifikasi citra penyakit daun kentang menggunakan CNN serta melakukan pengujian dengan data citra yang sudah di augmentasi.

1.5 Manfaat Penelitian

1. Memberi alternatif untuk mengetahui apakah daun kentang yang dimiliki berpenyakit atau tidak menggunakan algoritma CNN
2. Menjadi acuan dalam pengembangan pengklasifikasian secara otomatis jenis penyakit pada daun kentang

1.6 Tahapan Penelitian

Adapun tahapan penelitian yang dilakukan dalam menyelesaikan masalah penelitian adalah kuantitatif tipe ekperimental dengan dataset sekunder. Pengembangan perangkat lunak dilakukan dengan menggunakan metode *Linear Sequential Model* yang memiliki proses sistematis dengan pendekatan sekuensial (Widiyanto, 2018).Tahapan yang dilakukan sebagai berikut :

1. Pengumpulan data (akuisisi data)
2. Pemisahan data
3. Preprocessing data
4. Pemodelan Arsitektur menggunakan metode CNN (Convolutional Neural Network)
5. Pengujian Model
6. Analisis Pengujian
7. Perancangan Sistem

1.7 Sistematika Penulisan

Untuk mendapatkan gambaran dalam penulisan Tugas Akhir ini, maka secara garis besar sistematika dalam penulisan terbagi menjadi 5 (Lima) bab yaitu :

BAB I Pendahuluan

Pada bab ini berisi mengenai latar belakang, rumusan masalah, batasan masalah, tujuan penelitian,manfaat penelitian, tahapan penelitian, serta sistematika penulisan.

BAB II Tinjauan Literature

Pada bab ini membahas tentang penelitian serupa yang pernah dilakukan sebelumnya.

BAB III Metodologi Penelitian

Pada bab ini menjelaskan mengenai metode penelitian yang akan digunakan, pengumpulan data,metode yang diusulkan, tahapan penelitian, dan rancangan pengujian

BAB IV Hasil Dan Pembahasan

Dalam bab ini membahas mengenai implementasi Dan pembahasan program aplikasi yang telah disusun.

BAB V Penutup

Pada bab ini merupakan bagian akhir dari proposal penelitian, dimana di dalamnya terdapat kesimpulan serta saran pengembangan dari penelitian yang telah dilakukan untuk kedepannya

BAB II TINJAUAN LITERATUR

2.1 Tanaman Kentang

Tanaman kentang sangat bermanfaat dalam kehidupan manusia. Kentang merupakan salah satu makanan pokok karena mengandung karbohidrat. Tidak dapat dipungkiri bahwa kentang juga memiliki penyakit. Jika penyakit tanaman ini dibiarkan akan meningkatkan penurunan produksi pangan, maka perlunya penyakit dideteksi secara tepat waktu agar dapat dikendalikan secara efektif (Chen J. et al. 2020). Penyakit pada tanaman kentang yang paling sering ditemui adalah bercak kering daun (*early blight*), dan penyakit busuk daun (*late blight*). Tempat yang dingin dan lembab adalah salah satu faktor berkembangnya penyakit busuk daun (Fitriana et al. 2019).

2.2 Identifikasi Penyakit Daun Kentang

Ada 2 jenis penyakit yang umum menyerang tanaman kentang,yaitu bercak kering daun (*early blight*) dan busuk daun (*late blight*).Penyakit ini disebabkan oleh jamur, didukung oleh kondisi cuaca di lingkungan tumbuh.

Gambar 2.1 Early Blight

Penyakit ini disebabkan oleh jamur *Alternalia Solani*. Jamur ini biasanya muncul pada saat musim hujan. Penyakit ini dapat dikenali melalui daun dengan melihat bercak kecil berwarna coklat yang dapat membesar dan menyebabkan lubang pada daun.Bercak-bercak kecil ini membesar hingga diameter 3/8 inci secara bertahap. Bercak ini muncul sekitar dua sampai tiga hari setelah infeksi, dengan sporulasi lebih lanjut menyebar keseluruh daun dalam tiga sampai lima hari kemudian (Bauske et al., 2018).

2.2.2 Late Blight

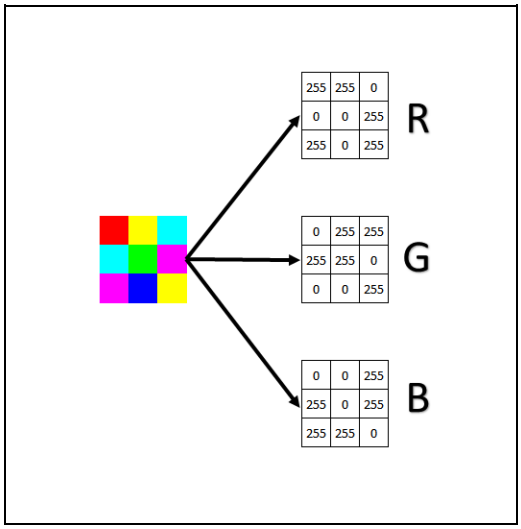
Penyakit ini ditandai dengan bercak coklat hingga hitam. Bercak akan muncul di ujung dan kemudian menyebar ke seluruh daun. Penyakit ini disebabkan oleh jamur *Phytophthora infestan*. Penyebaran virus ini sangat aktif,terutama saat musim hujan yang dingin dan lembab.Penyakit ini biasanya menyerang pada tanaman kentang yang berada di dataran tinggi.Bercak muncul pertama kali pada daun bagian bawah,kemudian mulai

berkembang didekat ujung atau tepi daun yang biasanya merupakan tempat ditemukannya embun (Robinson, 2017).

Gambar 2.2 Late Blight

2.3 Citra Digital

Citra digital merupakan salah satu jenis citra. Citra digital adalah representasi numerik dari data citra yang dapat diolah. Suatu citra digital dapat direpresentasikan dalam bentuk matriks menggunakan fungsi f(x,y) yang terdiri dari M kolom dan N baris. Perpotongan antara baris dan kolom disebut pixel. Satu pixel mewakili satu warna dan merupakan elemen terkecil dari citra digital (Gonzalez et al, 2002).Sebelum data diolah menggunakan convolutional neural network, citra terlebih dahulu direpresentasikan ke dalam bentuk numerik (citra digital).



Gambar 2.3 Representasi Citra RGB ukuran 3x3

Pada gambar 2.3 diilustrasikan hasil representasi citra RGB dengan ukuran 3x3. Hasil representasinya akan dimuat sebagai matriks. Pada kolom pertama baris pertama, warna merah direpresentasikan sebagai (255,0,0). Kemudian pada baris kedua kolom kedua warna kuning direpresentasikan sebagai (255,255,0).Pada citra RGB, representasi yang dihasilkan memiliki 3 lapisan matriks, masing-masing matriks merepresentasikan citra *red*, *green*, dan *blue*.

2.4 Pengolahan Citra (Image Processing)

Pengolahan citra atau *image processing* adalah sebuah proses mengolah piksel dalam citra digital untuk tujuan tertentu. Citra sering kali mengandung kesalahan (*noise*), warna yang terlalu kontras, tidak tajam, atau buram (*blurring*) dan sebagainya, sehingga menghasilkan kualitas yang buruk (degradasi). Tentu saja dengan citra yang seperti itu lebih sulit untuk di interpretasikan karena informasi yang disampaikan oleh citra tersebut menjadi berkurang. Dengan permasalahan ini maka citra perlu dimanipulasi. Bidang studi yang membahas hal ini adalah *image processing* (Munir, 2004).

Terdapat dua kelompok dalam pengolahan citra yaitu citra diperbaiki sesuai dengan kebutuhan dan mengolah informasi yang terdapat pada citra. Secara umum pengolahan informasi dalam suatu citra meliputi pengolahan citra dengan mengekstraksi informasi penting darinya. Dengan melakukan pengolahan citra diharapkan dapat diperoleh cirinya (Ahmad, 2005).

2.4.1 Operasi Pengolahan Citra Digital

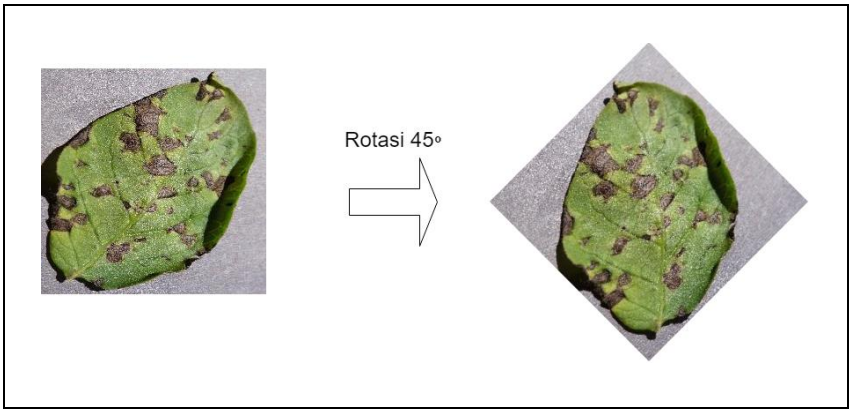
Terdapat berbagai operasi yang digunakan dalam mengolah citra pada penelitian ini, diantaranya adalah :

a. Resizing

Resizing merupakan suatu proses untuk mengubah resolusi dari sebuah citra, baik itu memperbesar ataupun memperkecil ukuran. Secara umum, dibutuhkan pemetaan dari sumber citra ke citra resized yang bertujuan untuk menjadikan citra menjadi semulus mungkin.

b. Rotasi (*Rotating*)

Rotasi adalah transformasi geometrik dimana nilai piksel suatu citra akan mengalami perubahan posisi dari yang didasarkan pada nilai variabel rotasi sebesar θ◦ terhadap posisi titik pusat rotasi. Rotasi dilakukan menggunakan Persamaan 2.1 dan dapat dilihat pada Gambar 2.4

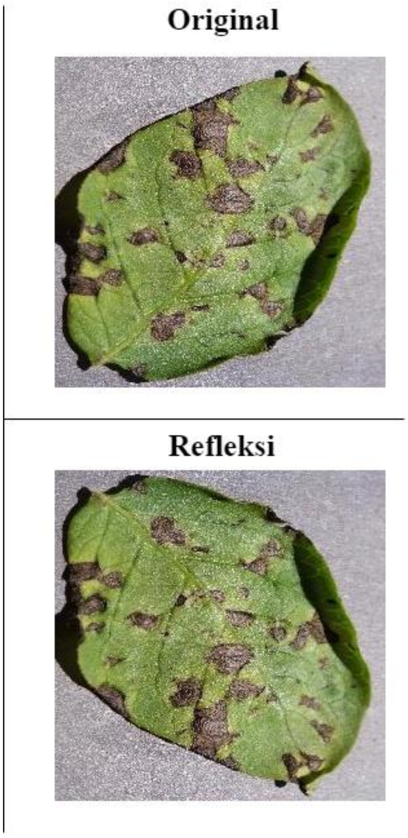


Gambar 2.4 Rotasi Sebesar 45°

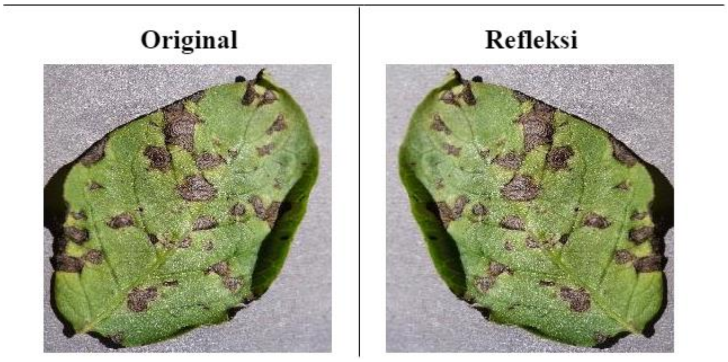
$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} \cos\theta^\circ & -\sin\theta^\circ \\ \sin\theta^\circ & \cos\theta^\circ \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} \quad \ldots\ldots (2.1)$$

c. Pencerminan (*flipping*)

Refleksi atau pencerminan adalah perubahan posisi dari nilai piksel berdasarkan koordinat awal (x1, y1) menuju koordinat akhir (x2, y2) pada citra sesuai dengan posisi pencerminan, ada dua jenis posisi pencerminan yang umum diterapkan yaitu pencerminan terhadap sumbu x dan pencerminan terhadap sumbu y. Pencerminan pada sumbu x dapat dilihat pada Gambar 2.5 dan pencerminan pada sumbu y dapat dilihat pada Gambar 2.6



Gambar 2.5 Perncerminan terhadap sumbu X



Gambar 2.6 Pencerminan terhadap sumbu Y

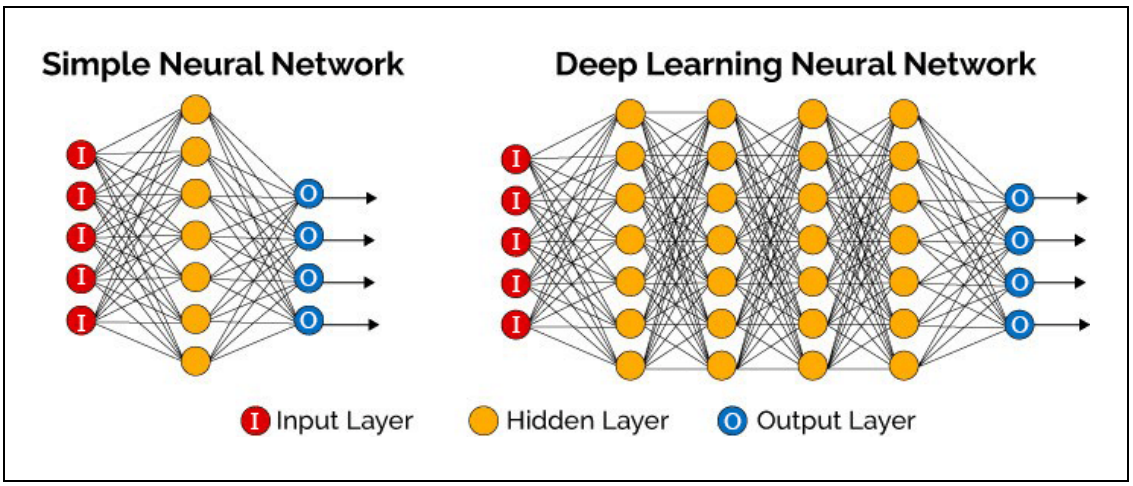
d. Min-Max

Min-max normalization merupakan proses untuk mengubah nilai dari sebuah pixel menjadi skala 0-255, sehingga dataset citra kentang memiliki skala yang sama setiap pixel nya. Proses ini dihitung dengan persamaan pada 2.2 berikut (Patro et al, 2015) :

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}} \quad \ldots\ldots (2.2)$$

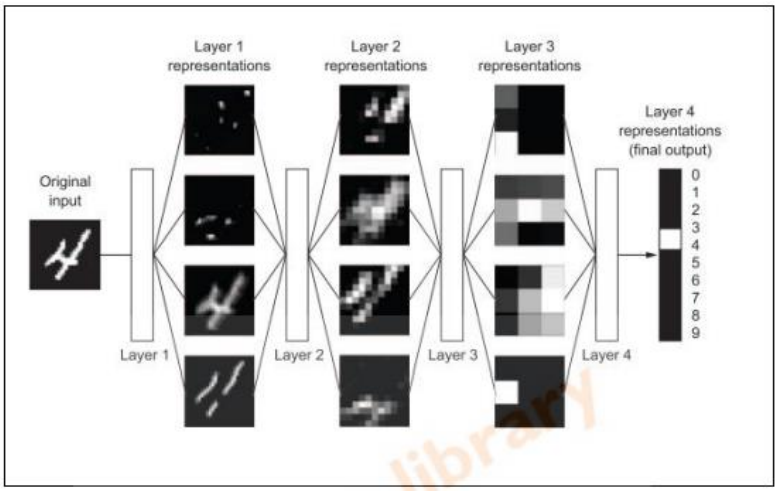
2.5 Deep Learning

Deep Learning merupakan bagian dari *machine learning* dengan banyak lapisan untuk membentuk tumpukan. Lapisan ini adalah algoritma atau metode yang digunakan untuk memproses data sehingga mendapatkan hasil terbaik (Nurfita and Ariyanto, 2018). Metode ini cocok digunakan ketika menangani masalah yang kompleks atau masalah dengan jumlah data yang besar. Akibatnya, proses pelatihan dalam metode *deep learning* akan membutuhkan waktu yang lama. Perbedaan *deep learning neural network* dan *simple neural network* dapat dilihat pada Gambar 2.7



Gambar 2.7 Perbandingan layer *machine learning* (kiri) dan *deep learning* (kanan)

Penerapan *deep learning* telah digunakan di beberapa bidang seperti klasifikasi video, klasifikasi gambar, *object detection*, *object recognition*,*natural language processing*,*robotic*,*text classification*, dan *singing synthetis* (Schmidhuber, 2015) Gambar 2.8 menjelaskan sebagian kecil proses klasifikasi gambar.

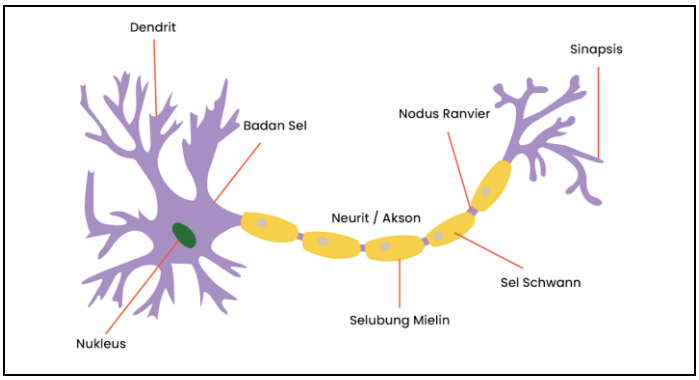


Gambar 2.8 Proses Training *Deep Learning*

2.6 Artificial Neural Network

Artificial Neural Network (ANN) adalah sistem cerdas yang digunakan untuk memproses informasi yang merupakan pengembangan dari model matematika umum. Prinsip kerja ANN terinspirasi dari prinsip kerja sistem jaringan saraf (neural network)

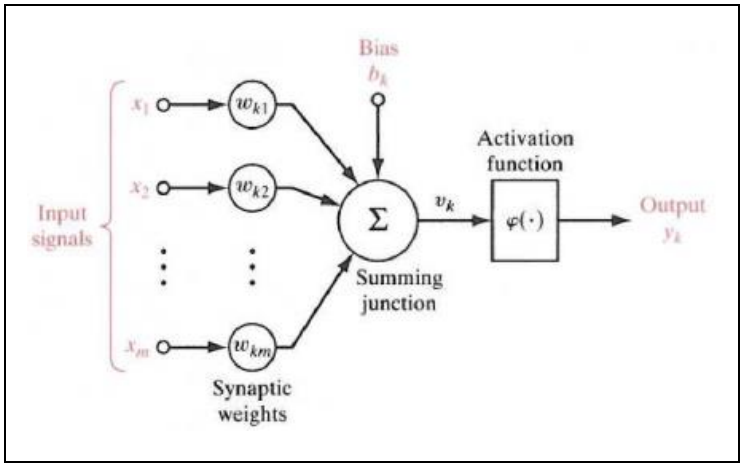
manusia (Gershenson, 2003). Jaringan Syaraf Tiruan (JST) menggunakan cara kerja otak manusia dengan tujuan meniru kecerdasan manusia. Otak manusia memproses informasi yang diterima dari indera manusia. Pemrosesan dilakukan oleh neuron yang terhubung antara jaringan syaraf dan bertindak berdasarkan sinyal listrik yang melewatinya. Jaringan syaraf menerapkan logika yang membuka dan menutup gerbang untuk mengirimkan sinyal listrik tergantung pada objek yang diterima dari alat indera (Giuseppe Ciaburro, 2017).



Gambar 2.9 Jaringan Syaraf Manusia

Karena ANN bekerja seperti jaringan syaraf biologi, maka bisa dikatakan bahwa ANN merupakan suatu generalisasi matematis dari cara kerja jaringan syaraf manusia.Cara kerja dari sebuah Neural Networks sebagai berikut:

a. Pemrosesan informasi terjadi pada neuron.
b. Sinyal mengalis diantara sel syaraf melalui suatu node.
c. Setiap node memiliki bobot yang bersesuaian. Bobot ini digunakan untuk menggandakan sinyal yang dikirim oleh neuron sebelumnya.
d. Setiap neuron menerapkan fungsi aktivasi terhadap hasil jumlah sinyal dengan bobot untuk menentukan sinyal output dari neuron tersebut dan disalurkan ke node dan neuron berikunya.



Gambar 2.10 Jaringan Syaraf Tiruan (JST)

Setiap *neuron* menerima input dan melakukan operasi dot pada sebuah *weight*,menjumlahkannya (*weighted sum*) dan menambahkan bias. Hasil dari operasi ini digunakan sebagai parameter dari *activation function* yang akan dijadikan output dari *neuron*

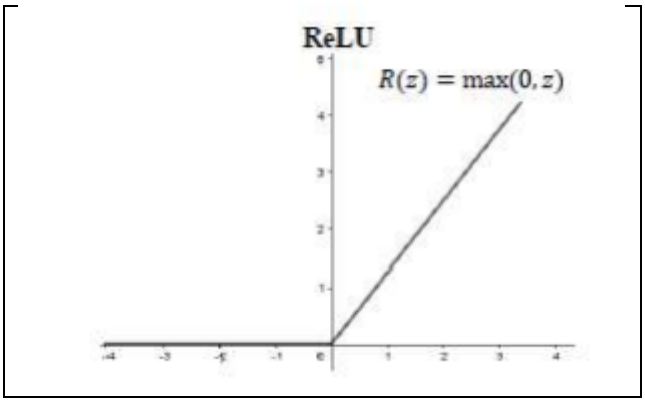
tersebut. *Activation function* digunakan untuk menentukan apakah sebuah neuron tersebut harus "aktif" atau tidak berdasarkan dari *weighted sum* dari input. *Output* neuron akan dimasukkan kedalam fungsi aktivasi, tanpa fungsi aktivasi output bisa berupa apa saja pada rentang [-*infinity*, +*infinity*] (Feng et al, 2019). Fungsi aktivasi yang digunakan pada penelitian ini antara lain :

2.6.1 ReLU Function

ReLU (*Rectified Linear Unit*) memiliki nilai berkisar dari nol sampai tak terhingga. Dapat dikatakan bahwa fungsi aktivasi ini menutupi kekurangan dari fungsi aktivasi lainnya. Fungsi ini melakukan *Thresholding* dengan nilai nol terhadap nilai piksel pada input citra. Fungsi aktivasi ReLU menghilangkan semua nilai piksel negatif pada suatu citra menjadi nol dan mempertahankan nilai yang positif. *Rectified Linear Unit* (ReLU) merupakan salah satu fungsi aktivasi yang banyak digunakan pada *Convolutional Neural Network* (Chen, Sun, & Wang, 2018). Bentuk fungsi ReLU :

Representasi fungsi aktifasi ReLU ditulisakan dalam persamaan Persamaan 2. 3

$$ReLu(a) = \max(0, a) \quad \text{..... (2.3)}$$



Gambar 2.11 Grafik fungsi ReLU

Kelebihan fungsi ReLU yaitu:

1. Fungsi ReLU konvergen terhadap stochastic gradient descent dibandingkan dengan fungsi sigmoid/tanh.
2. Operasi neuron pada fungsi ReLU lebih ringan dibanding fungsi sigmoid/tanh yang melibatkan operasi eksponensial. ReLU hanya melakukan treshholding sebuah matriks aktivasi pada nilai 0 (Chen, Sun, & Wang, 2018).
3. Melatih jaringan lebih cepat sehingga mengurangi kemungkinan Overfitting.

2.6.2 Softmax Function

Softmax Function adalah fungsi aktivasi yang melakukan normalisasi dari vektor suatu nilai tertentu kedalam distribusi probabilitas. *Softmax* memiliki nilai antara 0 dan 1.*Softmax* dapat digunakan untuk menentukan seberapa yakin model dalam mengenali image dan melakukan klasifikasi. Fungsi aktivasi *softmax* ada pada lapisan yang menghubungkan antara *Fully-Connected Layer* dengan *Dense Connection*. *Softmax* didefinisikan dalam persamaan 2.4

$$\sigma i(a)_i = \frac{e^a i}{\sum_{j=1}^{k} e^a j} \quad \text{....} \quad (2.4)$$

2.6.3 Loss Function

Goodfellow et al. (2016) menjelaskan *loss function* atau *cost function* adalah metode untuk mengevaluasi seberapa baik kinerja suatu algoritma dalam memodelkan kumpulan data tertentu.*Loss function* memiliki nilai dalam jumlah yang sangat besar ketika hasil prediksi menyimpang terlalu banyak dari hasil aktual. Secara bertahap, *loss function* belajar untuk mengurangi kesalahan prediksi dengan bantuan beberapa fungsi optimasi.

2.7 Optimizer

Algoritma optimasi bertujuan untuk meminimalkan kesalahan, menemukan bobot optimal dan memaksimalkan tingkat akurasi. Selama proses pelatihan, parameter (bobot) model diubah untuk meminimalkan *loss function* agar dapat memprediksi dengan sangat akurat. Tetapi permasalahnnya adalah kapan dan berapa banyak perubahan yang dilakukan dan bagaimana cara yang tepat masih belum dapat dipastikan, maka pada saat inilah pengoptimal masuk. Mereka menyatukan parameter model dan *loss function* dengan memperbaharui model sesuai dengan output dari fungsi kerugian.Sederhananya, pengoptimal menggunakan bobot untuk membentuk model menjadi bentuk yang paling akurat.Penelitian ini menggunakan jenis Adam,SGD,dan RMSProp.

2.7.1 RMSProp

Root Mean Square Propagation atau RMSprop adalah versi khusus Adagrad yang dikembangkan oleh Profesor Geoffrey Hinton di kelas jaringan sarafnya. Alih-alih mengumpulkan semua gradien terakumulasi untuk momentum, itu hanya mengakumulasi gradien di jendela tetap. RMSprop mirip dengan Adaprop, yang merupakan pengoptimal lain yang berupaya menyelesaikan beberapa masalah yang dibiarkan terbuka oleh Adagrad. RMSprop adalah pengoptimal yang memanfaatkan besarnya gradien terbaru untuk menormalkan gradient, yang menjaga rata-rata bergerak di atas gradien *root mean square* oleh karena itu disebut dengan Rms. $f'\ (\theta t)$ menjadi turunan dari loss sehubungan dengan parameter pada langkah waktu t. Dalam bentuk dasarnya, diberikan tingkat langkah α dan istilah peluruhan γ dilakukan pembaruan berikut:

$$r_t = (1-\gamma) f'\ (\theta_t)^2 + \gamma r_t - 1 \quad \text{....} \quad (2.5)$$

$$v_{\ t+1} = \frac{\alpha}{\sqrt{r_t}} f'\ (\theta_t) \quad \text{....} \quad (2.6)$$

$$\theta_t + 1 = \theta_t - v_{t+1} \quad \text{....} \quad (2.7)$$

Keterangan:
α : tingkat pembelajaran awal (learning rate)
γt : rata-rata eksponensial dari kotak gradien
θt : gradien pada waktu t sepanjang v

2.7.2 Stochastic Gradient Descent (SGD) optimizer

Gradient Descent adalah salah satu algoritma yang paling populer untuk mengoptimalkan model jaringan syaraf tiruan. Algoritma ini juga merupakan metode yang

banyak digunakan dalam berbagai macam model pembelajaran. Pada dasarnya, ketika melatih sebuah model dibutuhkan sebuah *Loss Function* yang dapat mengukur kualitas tiap bobot atau parameter tertentu. Tujuan dari melakukan optimasi adalah untuk menemukan parameter yang dapat meminimalkan *Loss Function*.

Fungsi objektif meningkat secara tidak beraturan karena SGD sering diperbaharui dengan varians yang tinggi. Hal tersebut dapat membuat *Loss Function* melompat ke titik minimal yang baru dan menimbulkan potensi melompat ke minimum yang tidak pasti. Namun, hal ini dapat dicegah dengan menurunkan *Learning Rate* dan SGD akan menuruni *Loss Function* ke titik minimum dengan optimal.

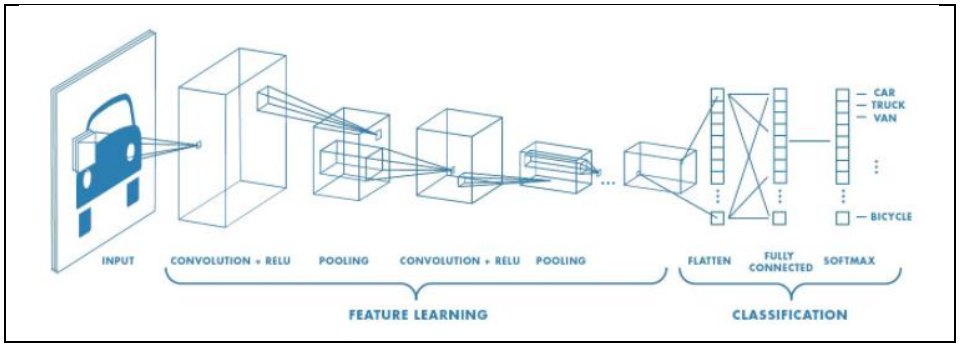
2.7.3 Adam Optimizer

Adam adalah algoritma pengoptimalan yang dapat digunakan sebagai ganti dari prosedur stochastic gradient descent klasik untuk memperbarui *weight network* secara iteratif berdasarkan data training. Adam adalah algoritma yang populer di bidang deep learning karena ia mencapai hasil yang baik dengan cepat.Hasil empiris menunjukkan bahwa Adam bekerja dengan baik dalam praktiknya dan lebih baik dibandingkan dengan stochastic optimization method lainnya (Brownlee 2017).

2.8 Convolutional Neural Network

Convolutional Neural Network (CNN) merupakan salah satu pengembangan lebih lanjut dari *Artificial Neural Network* (ANN). CNN merupakan algoritma *neural network* yang digunakan untuk input berupa gambar atau video (Santosa & Umam, 2018). Salah satu keunggulan dari CNN yaitu mengurangi jumlah parameter dalam *Neural Network* (Albawi, Mohammed, & Al-Zawi, 2017).

Secara umum, arsitektur CNN terdiri dari 2 tahap. Pertama adalah tahap ekstraksi fitur (*feature extraction / feature learning*). Tahap ini biasanya mencakup operasi konvolusi, fungsi aktivasi (ReLU), dan *pooling*. Pada tahap ini, parameter yang dihasilkan berupa angka-angka yang mewakili hasil operasi yang dilakukan. Kedua, tahap klasifikasi dimana hasil dari ekstraksi fitur diklasifikasikan melalui *full connected layer*.

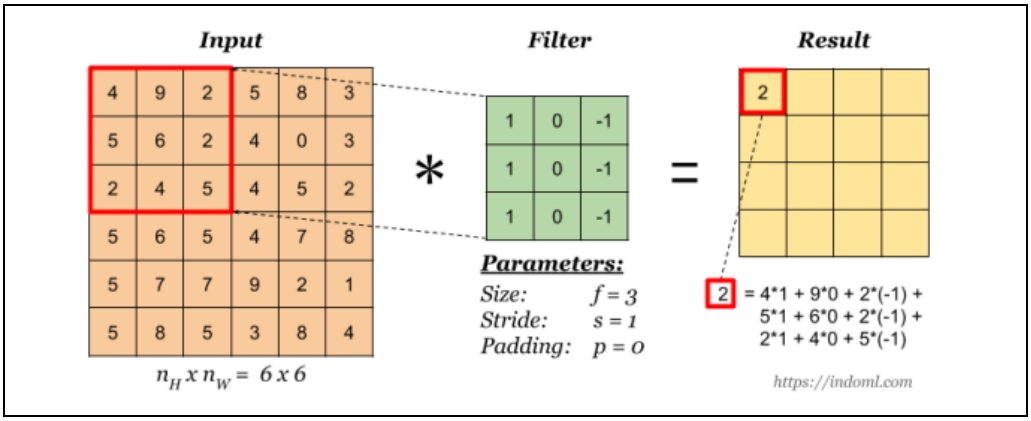


Gambar 2.12 Contoh Arsitektur CNN

2.8.1 Convolutional layer

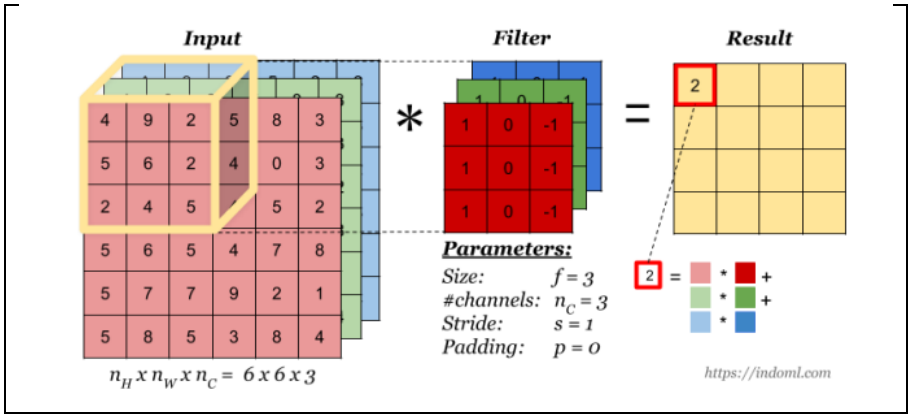
Lapisan konvolusional (*convolutional layer*) adalah lapisan inti CNN,dimana sebagian besar proses komputasi dilakukan. Tujuan utama konvolusi dalam terminologi dengan ConvNet adalah untuk mengekstraksi fitur dari citra yang dimasukkan. Lapisan

konvolusi terdiri dari atas struktur dengan sejumlah filter dengan ukuran tetap yang memungkinkan fungsi kompleks diterapkan pada gambar yang telah dimasukkan. Proses ini dilakukan dengan cara menggeser filter di atas citra. Selama proses ini, setiap filter memiliki bobot dan nilai bias yang sama di seluruh citra. Proses ini dikenal sebagai mekanisme pembagian bobot dan memungkinkan fitur yang sama direpresentasikan di seluruh citra (Sarigul, et al. 2019). Lapisan konvolusi menghasilkan gambar baru yang disebut peta fitur.Peta fitur menonjolkan fitur unik dari gambar asli.Lapisan konvolusi bekerja dengan cara yang sangat berbeda dari lapisan jaringan saraf lainnya. Lapisan konvolusi menghasilkan peta fitur sebanyak filter konvolusi. Sehingga, jika lapisan konvolusi berisi empat filter,dihasilkan empat peta fitur (Kim,2017).



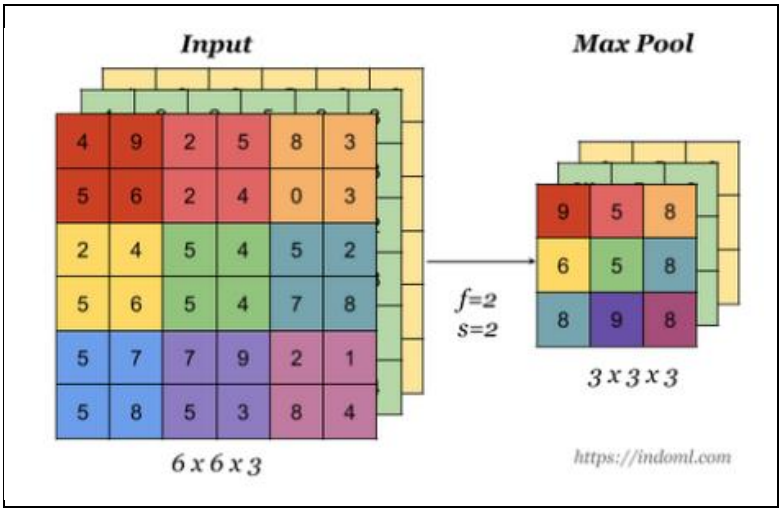
Gambar 2.13 Ilustrasi operasi konvolusi

Operasi konvolusi pada Gambar 2.13 hanya berlaku untuk citra *grayscale*, dan representasi gambar yang dihasilkan hanya satu layer. Untuk citra RGB, representasi gambat yang dihasilkan sebanyak tiga lapisan matriks. Ketiga lapisan matriks ini hanya akan menghasilkan satu lapis matriks dari operasi konvolusi. Masing-masing lapisan matriks akan dioperasikan dengan *filter*. Hasil operasi konvolusi setiap layer ditambahkan untuk menghasilkan matriks konvolusi,seperti yang ditunjukkan pada Gambar 2.14 berikut.



Gambar 2.14 Ilustrasi operasi konvolusi pada citra RGB

Tujuan utama dari *pooling* adalah untuk mengurangi kompleksitas layer berikutnya. Dalam hal citra, *pooling* akan mengurangi resolusi gambar. Ini mengurangi jumlah parameter yang perlu diperbarui, mempercepat komputasi, dan menghindari *overfitting*. *Pooling* tidak mempengaruhi jumlah layer. Salah satu jenis *pooling* yang paling umum digunakan yaitu *max pooling* (Albawi, Mohammed, & Al-Zawi, 2017).

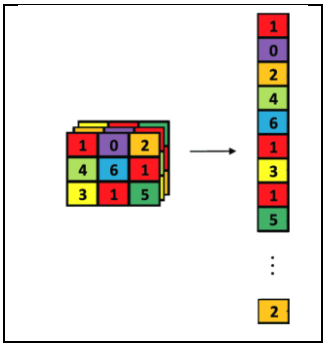


Gambar 2.15 Ilustrasi operasi *pooling*

Pada operasi *pooling*, ada 2 argumen yang perlu diperhatikan, yaitu *filter* dan *stride*. *Filter* adalah ukuran matriks yang digunakan untuk melakukan *pooling*. *Stride* merupakan jumah kolom dan/atau baris yang dipindahkan setelah melakukan satu operasi *pooling*. Misalnya pada gambar 2.15, filter yang digunakan adalah 3x3 dengan *stride* sebanyak 2. Operasi yang digunakan yaitu *max pooling*. Operasi dilakukan dari kiri atas ke kanan bawah. Hasil dari setiap operasi *max pooling* merupakan nilai maksimum dari lingkup *filternya*. Operasi pertama memiliki nilai 4, 9, 5, dan 6 di area filter. Hasil dari operasi tersebut yaitu 9 yang merupakan nilai tertinggi dari empat nilai lainnya.

2.8.2 Flatten

Flatten merupakan metode transformasi data matriks n-dimensi menjadi satu dimensi. *Flatten* ini digunakan setelah *hasil* hasil akhir ekstraksi fitur. *Output* berupa matriks n-dimensi diubah menjadi 1 dimensi untuk klasifikasi pada *fully connected layer*.



Gambar 2.16 Ilustrasi *Flatterning*

2.8.3 Fully Connected Layer

Fully connected layer merupakan bagian dari CNN dan menyerupai neural network biasa. Setiap node dalam layer ini saling terhubung langsung dengan setiap node pada layer sebelumnya dan selanjutnya. *Fully connected layer* ini merupakan lapisan yang paling banyak menggunakan parameter dan membutuhkan waktu komputasi yang tinggi saat *training* (Albawi, Mohammed, & Al-Zawi, 2017). Pada dasarnya layer ini biasanya digunakan pada MLP (*Multi Layer Perceptron*) yang tujuannya untuk mentransformasikan

dimensi data sehingga data dapat diklasifikasikan secara linear.Input layer pada FC merupakan hasil *flattening* seperti pada gambar 2.16.*Fully Connected lay*er di definisikan dalam persamaan 2.5 berikut :

$$y_i = \Sigma_j W_{ij} X_j \quad \text{......} \quad (2.5)$$

2.9 Hyperparameter

Hyperparameter merupakan parameter yang perlu untuk diatur sebelum *training* di CNN. *Variable* dalam *hyperparameter* menentukan bagaimana sebuah jaringan dilatih (Rima Dias Ramadhani et al., 2021). Pengaturan *hyperparameter* ditujukan untuk mengoptimalkan bobot dan bias. Contoh *hyperparameter* adalah *Learning rate* yang menentukan seberapa cepat jaringan memperbarui bobot dan biasnya. *Epoch* adalah proses iterasi pada dataset yang berulang setelah proses training. *Stride* adalah nilai ukuran pergeserain yang digunakan untuk menggeser *filter*. *Padding* merupakan penambahan piksel bernilai 0 pada bingkai luar citra.

2.10 Confusion Matrix

Confusion matrix adalah salah satu metode pengukuran keputusan yang paling banyak digunakan dalam *supervised machine learning. Confusion matrix* memvisualisasikan nilai tingkat kebingungan dari algoritma pada setiap kelas yang berbeda dan tidak tergantung pada algoritma klasifikasi.Tujuan dari *confusion matrix* adalah untuk melakukan perhitungan akurasi pada konsep data mining.Akurasi klasifikasi adalah persentase ketepatan *record* data yang diklasifikasikan dengan benar setelah dilakukan pengujian pada hasil klasifikasi (Mayadewi et al., 2015).

Tabel 2.1 *Confusion Matrix*

Nilai Prediksi	**Nilai Sebenarnya**	
	TRUE	**FALSE**
TRUE	True Positive (A)	False Positive (B)
FALSE	True Negative (C)	True Negative (D)

Keterangan :

1. True Negative (A) : jumlah data set negatif yang diklasifikasikan sebagai negatif juga.
2. False Positive (B): jumlah data set negatif yang diklasifikasikan sebagai positif.
3. False Negative (C): jumlah data set positif yang diklasifikasikan sebagai negatif.
4. True Positive (D) : jumlah data set positif yang diklasifikasikan sebagai positif juga.

Perhitungan akurasi dengan tabel *confusion matrix* adalah sebagai berikut :

$$\text{Akurasi} = \frac{(A+D)}{(A+B+C+D)} \quad \text{......} \quad (2.6)$$

2.11 Penelitian Sebelumnya

Sebuah penelitian yang berjudul Klasifikasi Penyakit Daun Kentang Berdasarkan Fitur Tekstur Dan Fitur Warna Menggunakan Su*pport Vector Machine* (Rakhmawati et al., 2018b) bertujuan untuk mengidentifikasi penyakit pada tanaman kentang menggunakan metode *Support Vector Machine* (SVM).Penelitian dilakukan dengan dataset didapatkan dari Badan Pengkajian Teknologi Pertanian.Database image ini berjumlah masing–masing 100 citra early, 100 citra blight, dan 100 citra non disease / normal. Metode ekstraksi fitur tekstur yang digunakan adalah Grel Level Co-occurrence matrix dengan fitur ciri, sedangkan fitur warna yang digunakan adalah *Color Moments* dari masing–masing *channel* HSV. Pengenalan penyakit daun kentang ini akan diklasifikasikan ke tiga kelas yaitu *early blight,late blight* dan *undisease/*normal menggunakan metode SVM. Pada penelitian ini, akurasi tertinggi yang dicapai adalah 90%, tetapi kinerja dalam membedakan daun *non disease* atau normal mengalami penurunan sebesar 83,33%.

Dalam penelitian lainnya yang berjudul *Early detection and classification of plant diseases with Support Vector Machines based on hyperspectral reflectance* (Rumpf et al., 2010) bertujuan untuk pendeteksian dini penyakit bercak daun Cercospora, karat daun dan embun tepung pada daun bit gula dengan menggunakan pencitraan hyperspectral. Deteksi dini penyakit pada tanaman sangat penting untuk perlindungan tanaman yang presisi.Kontribusi utama dari paper ini adalah prosedur untuk deteksi dini dan diferensiasi gula bit penyakit berdasarkan Support Vector Machines dan indeks vegetasi spektral.Peneliti menggunakan algoritma SVM untuk melakukan pekerjaan klasifikasi. Hasil dari penelitian ini,algoritma SVM mampu mengklasifikasikan 3 jenis penyakit dengan kondisi sehat mencapai akurasi diatas 86%.

Penelitian selanjutnya berjudul Deteksi Penyakit Tanaman Rambutan Berdasarkan Citra Daun Menggunakan *Fuzzy K-Nearest Neighbour* (Rahayu & Mendes, 2021) bertujuan untuk mengenali penyakit yang menyerang pada tanaman dengan menggunakan citra daun.Penelitian ini dilakukan menggunakan algoritma *Fuzzy K-Nearest Neighbour* dan memberikan hasil perkiraan untuk pengambilan keputusan pada kondisi daun itu sehat atau tidak. Aplikasi ini mengklasifikasi daun rambutan menjadi 4 kelas yaitu, kelas daun sehat, kelas penyakit embun jelaga, kelas hama kutu putih, dan kelas hama ulat.Klasifikasi penyakit yang dilakukan pada penelitian ini berdasarkan nilai RGB dan HSV citra dengan akurasi hasil adalah 67% dengan k=7.

Penelitian lain yang berjudul Perancangan dan Simulasi Deteksi Penyakit Tanaman Jagung Berbasis Pengolahan Citra Digital Menggunakan Metode Color Moments dan GLCM (Sari, 2016) melakukan klasifikasi menggunakan Metode KNN dengan ekstraksi ciri menggunakan metode Color Moments dan Gray Level Co-Occurrence Matrix (GLCM).Salah satu kendala penting dalam upaya peningkatan produksi jagung adalah gangguan biotis. Gejala penyakit bisa dilihat dari perubahan yang terjadi pada daun jagung.Pendeteksian penyakit pada daun tanaman jagung ini diproses melalui pengolahan sinyal digital dan terdiri dari 4 bagian utama, yaitu Preprocesssing, ekstraksi ciri warna, ekstraksi ciri tekstur, dan klasifikasi. Dataset diambil sendiri dengan jumlah 160 citra dengan distribusi 40 citra per penyakit.Dataset di *capture* dalam jarak ±15 cm. Kemudian citra tersebut disimpan di dalam PC sehingga dapat diakses melalui sistem. Setelah itu, citra

kemudian terlebih dahulu dilakukan proses resize menjadi 800 x 1000 pixel.Hasil pengujian menunjukkan rata- rata akurasi dari klasifikasi adalah 89.375%, menggunakan *Euclidean Distance* dimana nilai k = 1. Hasil ini merupakan rata-rata dari nilai ciri dari masing-masing parameter ciri.

Dalam penelitian lainnya yang berjudul *Blackleg Detection in Potato Plants using Convolutional Neural Networks* (D. Oppenheim et al.,2017) melakukan klasifikasi penyakit pada citra tanaman kentang menggunakan CNN. Banyak penyakit tanaman memiliki gejala visual yang berbeda yang dapat digunakan untuk mengidentifikasi dan mengklasifikasikannya dengan benar. Jurnal ini membuat algoritma klasifikasi penyakit kentang yang memanfaatkan penampilan berbeda kentang.Basis data gambar yang digunakan dalam penelitian ini, berisi 400 gambar kentang berbagai bentuk, ukuran dan penyakit, diperoleh, diklasifikasikan, dan diberi label secara manual oleh para ahli di bawah kondisi pencahayaan normal yang terkendali. CNN di training dengan berbagai ukuran set pelatihan.Terdapat 4 penyakit berbeda pada kentang yang dilihat dari kulitnya Hasil penelitian yang didapatkan mencapai tingkat akurasi 96% menguunakan asitektur jaringan VGG dengan sebaran data untuk latih 90% dan data uji 10%.Hasil yang didapatkan menunjukkan bahwa CNN terlatih dapat mengklasifikasikan dengan benar dan tepat keempat penyakit tersebut disajikan di sini.

Penelitian selanjutnya yang berjudul Tomato *Leaf Disease Detection using Convolution Neural Network* (M. Agarwal, A. Singh et al.,2019) juga melakukan identifikasi jenis penyakit yang menyerang tanaman tomat. Kualitas dan Kuantitas tanaman tomat menurun akibat berbagai macam penyakit,sehingga untuk mendeteksi dan mengklasifikasikan penyakit tersebut maka digunakan Convolution Neural Network (CNN).Dataset yang digunakan diambil dari kaggle yang merupakan 1000 gambar termasuk dalam kategori sehat dan 1000 gambar termasuk dalam setiap kategori penyakit tomat.Dalam set validasi, setiap kelas memiliki 700 gambar dan set pengujian memiliki 50 gambar di setiap kelas.Arsitektur yang diusulkan penulis kemudian dibandingkan dengan VGG16, Mobilnet dan InceptionV3 dengan menggunakan dataset penyakit daun tomat dari dataset PlantVillage. Hasil yang didapatkan yakni Akurasi pengujian rata-rata dari model yg dikembangkan adalah 91,2% ,lebih tinggi dari model mobilenet,VCG 16,dan Inception V3.Ruang penyimpanan yang dibutuhkan oleh model yang diusulkan adalah sebesar 1,5 MB sedangkan model sebelumnya memiliki kebutuhan ruang penyimpanan sekitar 100 MB sehingga menunjukkan manfaat model yang diusulkan dibandingkan model yang telah dilatih sebelumnya.

Penelitian selanjutnya berjudul Deteksi Penyakit Tanaman Cabai Menggunakan Metode Convolutional Neural Network (A. Tsany, R. Dzaky,2021) melakukan klasifikasi penyakit pada citra daun cabai. Penyakit dari suatu tanaman akan sangat mempengaruhi hasil panen dari tanaman tersebut. Jika penyakitnya tidak segera ditangani maka penyakit tersebut dapat merusak tanaman dan mengakibatkan gagal panen yang akan berpengaruh ekonomi. Dengan menggunakan image processing penyakit yang dapat dilihat dan direkam oleh kamera akan dapat dianalisis dan diidentifikasikan oleh komputer.Pada penelitian kali ini digunakan citra daun cabai yang diambil langsung oleh peneliti dari perkebunan cabai di Jawa Tengah.Data diambil sekitar pukul 10.00-12.00 siang sehingga mendapatkan citra

dengan warna yang jelas. Model ini akan bisa mengklasifikasikan 4 jenis kondisi daun dengan 3 penyakit dan 1 kondisi normal. Model CNN ini bisa menghasilkan akurasi 80% untuk epoch 10 dan 90% untuk epoch 40 menggunakan arsitektur AlexNet. Dengan pengaturan learning rate yang sesuai, maka model akan lebih cepat memahami citra dan memberikan akurasi yang cukup baik.

Penelitian selanjutnya berjudul *Varietal Classification Of Barley By Convolutional Neural Networks* (Kozłowski et al., 2019) melakukan klasifikasi pada biji tanaman gandum.Dataset diperoleh dari pertanian terpilih di wilayah Warmia-Mazury Polandia. Sampel terdiri dari 11 varietas jelai.Gambar berwarna diperoleh melalui pemindai flatbed Epson 4990 pada 400 dpi.Model CNN terdiri dari lapisan atau *layer* dengan ukuran 80x170 piksel dengan 3 saluran atau *channel* warna.Dengan menggunakan lapisan fungsi aktivasi ReLU dan diikuti *pooling layer* dengan ukuran 3x3. Akurasi terbaik yang didapatkan adalah 93,20%.

Penelitian selanjutnya berjudul *Hyperspectral fruit and vegetable classification using convolutional neural networks* (Steinbrener et al., 2019) melakukan klasifikasi menggunakan pendekatan pseudo-RGB yang memanfaatkan informasi tambahan yang disediakan oleh gambar hyperspectral untuk mencapai akurasi yang lebih baik bahkan untuk arsitektur CNN dengan data pelatihan (dataset) terbatas.Dataset yang digunakan adalah Data yang diambil sendiri dengan camera Ximea (MQ022HG-IM- SM4X4-VIS) merekam total 2700 gambar dari 13 kelas buah dan sayuran yang berbeda dengan ukuran 256x256 piksel,lalu *resize* menjadi 224x224 piksel yang kemudian dijadikan sebagai data input.Penelitian menggunakan *machine learning* GoogleNet sebagai dasar pada transfer learning,pada GoogleNet terdiri dari 9 modul *inception* serta beberapa lapisan konvolusional sederhana dan beberapa lapisan yang sepenuhnya terhubung (*fully connected layer*).Pada proses pelatihan menggunakan *learning rate* sebesar 0,0001 dengan batch sebesar 32.Hasil penelitian ini mencapai akurasi 92.33 %.

Penelitian lainnya berjudul Deteksi Penyakit pada Daun Kentang Menggunakan Metode *Convolutional Neural Network* (Rozaqi et al., 2021) melakukan klasifikasi menggunakan CNN untuk mendeteksi penyakit pada citra daun kentang.Berdasarkan implementasi model yang dilakukan dan pengujian yang dilakukan pada data daun kentang memiliki hasil yang bagus dengan pembagian dataset 80% dan 20% dan gambar yang digunakan ukuranya dirubah 150x150. Pada epoch ke 10 dengan batch_size 20 dengan total data training 922 gambar dan data testing 230 gambar menghasilkan nilai akurasi 95% dan untuk akurasi validasi menghasilkan 94%.

Penelitian yang bertujuan untuk mengidentifikasi tanaman telah dilakukan oleh banyak peneliti,namun teknologi selalu menghadirkan algoritma-algoritma terbaru dan kinerjanya meningkat dari tahun ke tahun .Dari semua penelitian yang ada algoritma CNN memiliki tingkat akurasi tertinggi dibandingkan dengan algoritma lain.Oleh karena itu penulis bermaksud untuk menggunakan algoritma CNN untuk Klasifikasi Penyakit pada Daun Kentang Menggunakan Pengolahan Citra.

Dari kesebelas penelitian sebelumnya,maka dapat diringkas menjadi suatu tabel *state of the art* yang dapat dilihat pada **Tabel 2.2** berikut ini :

Tabel 2.2 *State of The Art*

No	Topik/ Tema Penelitian	Objek Penelitian	Metode yang Digunakan	Sumber Data (Dataset)	Hasil
1.	Image Processing (T. Rumpf, U. Steiner,2010)	Deteksi dini penyakit bercak daun dengan menggunakan pencitraan hyperspectral	Method : *Support Vector Machine* (SVM)	Dataset didapatkan dari cv. Pauletta KWS GmbH Einbeck, Germany	Mampu mengklasifikasikan 3 jenis penyakit dengan kondisi sehat mencapai diatas 86%. Peneliti juga mendapatkan akurasi klasifikasi 65% hingga 90% tergantung stadium penyakitnya.
2.	Image Processing (M. Agarwal, A. Singh, S. Arjaria ,2019)	Deteksi penyakit tomat dengan citra daun menggunakan metode CNN	Metode : *Convolutional Neural Network* (CNN)	Data tersedia dan didapatkan di Kaggle dengan nama Plant Village dataset	Akurasi pengujian rata-rata dari model yg dikembangkan adalah 91,2% lebih tinggi dari model mobilenet,VCG 16,Inception V3. Ruang penyimpanan yang dibutuhkan oleh model yang diusulkan adalah sebesar 1,5 MB sedangkan model sebelumnya memiliki kebutuhan ruang penyimpanan

					sekitar 100 MB sehingga menunjukkan manfaat model yang diusulkan dibandingkan model yang telah dilatih sebelumnya.
3.	Image Processing (D. Oppenheim, G. Shani,2017)	Klasifikasi penyakit pada citra buah kentang menggunakan metode CNN	Method : *Convolutional Neural Network* (CNN)	Dataset berjumlah 400 gambar kentang berbagai bentuk yang diambil manual.	Menggunakan lebih banyak data untuk fase pelatihan meningkatkan klasifikasi dan mengurangi tingkat kesalahan.Hasil ini menunjukkan bahwa CNN terlatih dapat mengklasifikasikan dengan benar dan tepat keempat penyakit tersebut disajikan di sini.
4.	Image Processing (A. Tsany, R. Dzaky,2021)	Deteksi penyakit tanaman cabai dengan citra daun menggunakan metode CNN	Method : *Convolutional Neural Network* (CNN)	Dataset diambil langsung dari perkebunan cabai di Jawa Tengah	Penggunaan arsitektur AlexNet dalam penelitian ini juga menghasilkan akurasi yang tinggi. Akurasi yang dihasilkan 80% untuk epoch 10 dan 90% untuk epoch 40
5.	Image Processing (Intan Permata Sari, Bambang Hidayat, Ratri Dwi Atmaja, 2016)	Deteksi penyakit pada jagung dengan Metode Color Moments dan GLCM	Metode : *Color Moments* dan *Gray-Level Coo Occurrence Matrix* (GLCM)	Dataset diambil sendiri dengan jumlah 160 citra dengan distribusi 40 citra per penyakit	Pengujian menggunakan metode ekstraksi warna (Color Moment) dan ekstraksi tekstur (GLCM) mencapai akurasi tertinggi sebesar 89,375%
6.	Image Processing (Jan Steinbener, Konstantin	Klasifikasi buah dan sayuran	Metode : *Convolutional Neural*	Data diambil sendiri dengan	Menggunakan pendekatan pseudo-RGB yang memanfaatkan

	Posch, Raimund Leitner, 2019)	hiperspektral dengan CNN	*Network* (CNN)	camera Ximea (MQ022HG-IM-SM4X4-VIS) merekam total 2700 gambar dari 13 kelas buah dan sayuran yang berbeda.	informasi tambahan yang disediakan oleh gambar hyperspectral untuk mencapai akurasi yang lebih baik bahkan untuk arsitektur CNN dengan data pelatihan (dataset) terbatas.Akurasi yang didapatkan adalah 92.33 %
7.	Image Processing (Jan Steinbener, Konstantin Posch, Raimund Leitner, 2019)	Klasifikasi Varietas Biji Gandum Dengan Menggunakan Metode CNN	Metode : *Convolutional Neural Network* (CNN)	Biji-bijian gandum diperoleh dari pertanian terpilih di wilayah Warmia-Mazury Polandia. Sampel terdiri dari 11 varietas jelai.Gambar berwarna diperoleh melalui pemindai flatbed Epson 4990 pada 400 dpi.	Hasil penelitian ini menunjukkan bahwa pengenalan varietas kernel individu dapat dicapai oleh CNN dengan akurasi yang memuaskan. Jika dibandingkan dengan metode tercanggih, penerapan jaringan dalam memungkinkan peningkatan akurasi klasifikasi hingga lebih dari 40%. Hasil ini menunjukkan bahwa metode visi komputer dan jaringan saraf dalam CNN dapat berhasil diterapkan di industri malting untuk penilaian kualitas jelai.Akurasi yang didapat 93%.
8.	Image Processing (Guntur Wicaksono,Septi	Deteksi Penyakit pada Daun Tanaman Apel	Metode : *Convolutional Neural Network* (CNN)	Dataset yang berasal dari PlantVillage yang	Rata-rata akurasi model untuk data training epoch 50,75,100 adalah 99,2% dan memiliki

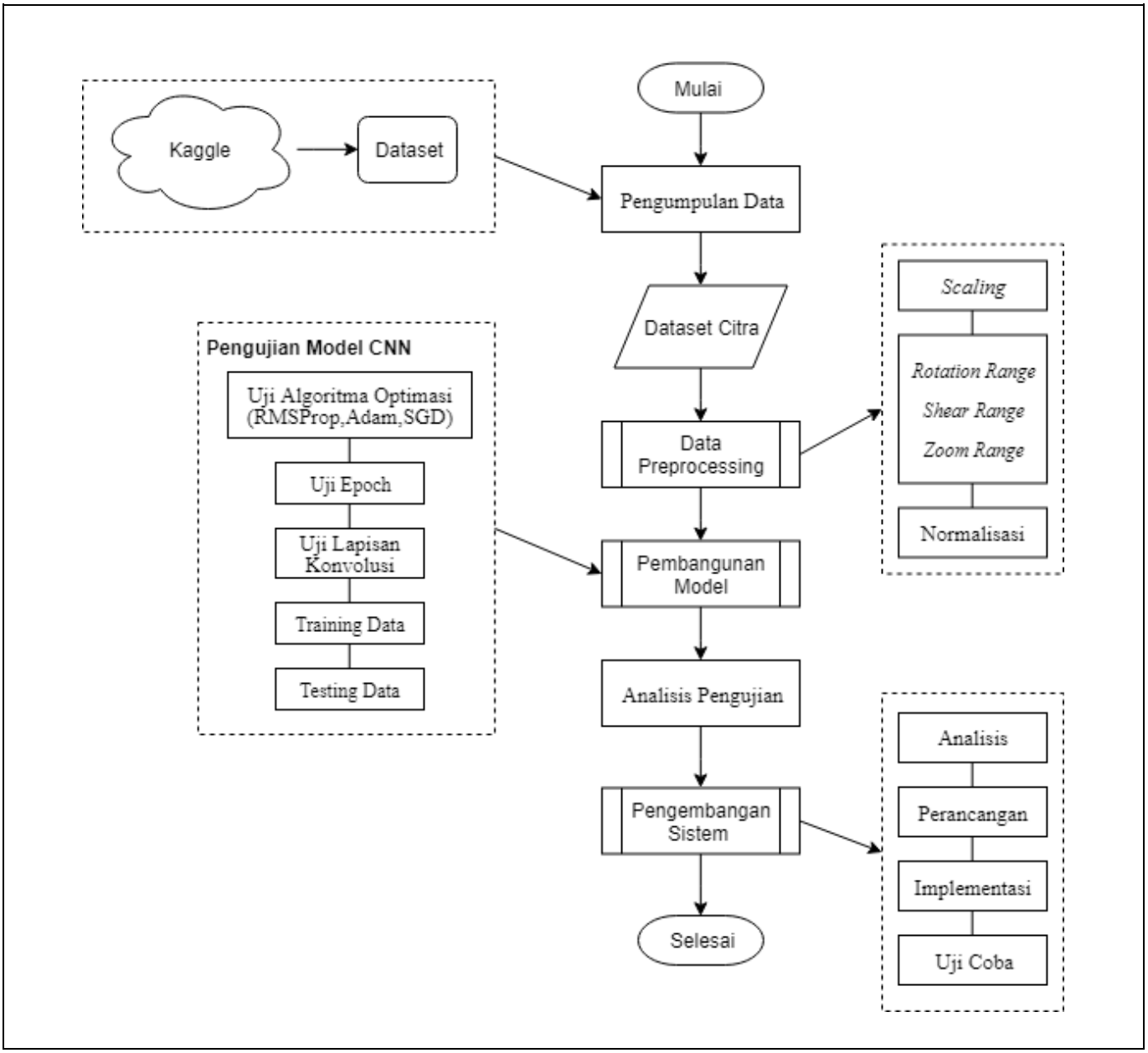
	Andryana,Benrahman, 2020)		dengan Transfer Learning LeNet-5	dibuat oleh SP Mohanty CEO & Co-founder CrowdAI dengan jumlah sebanyak 3151 citra daun yang telah diklasifikasi berdasarkan kelasnya masing-masing.	rata-rata loss model sebesar 0,063. Untuk data testing didapat hasil untuk epoch 50, 75 dan 100 dengan rata-rata akurasi sebesar 94,9% dan memiliki rata-rata loss validasi sebesar 0,277.
9.	Image Processing (Puji Utami Rakhmawati, Yuliana Melita Pranoto, Endang Setyati, 2018)	Klasifikasi Penyakit Daun Kentang Berdasarkan Fitur Tekstur Dan Fitur Warna	Metode : *Support Vector Machine* (SVM)	Dataset didapatkan dari Badan Pengkajian Teknologi Pertanian.Database image ini berjumlah masing – masing 100 citra early, 100 citra blight, dan 100 citra non disease / normal.	Pada penelitian ini, akurasi tertinggi dicapai dalam mengidentifikasi jenis penyakit early blight dan late blight, dengan nilai akurasi masing-masing 90%, tetapi kinerja dalam mengidentifikasi daun bebas penyakit atau normal mengalami penurunan sebesar 83,33%.
10.	Image Processing (Tri Kustanti Rahayu, Johana Anike Mendes, 2021)	Deteksi Penyakit Tanaman Rambutan Berdasarkan Citra Daun	Metode : *Fuzzy K-Nearest Neighbour* (FKNN)	Dataset tidak diberitahu penulis	Pada penelitian ini, peneliti menggunakan nilai RGB dan HSV dari citra untuk melakukan metode Fuzzy K Nearset Neighbor untuk klasifikasi penyakit dengan akurasi 67%.

11	Image Processing	Deteksi penyakit pada citra daun kentang	Metode : *Convolutional Neural Network* (CNN)	Dataset didapatkan dari kaggle	Nilai akurasi sebesar 95% untuk epoch ke-10 dengan batch_size 20, yang berisi total 922 citra data latih dan 230 citra uji data dan untuk akurasi validasi menghasilkan 94%.

BAB III METODOLOGI PENELITIAN DAN PENGEMBANGAN SISTEM

3.1 Metodologi Penelitian

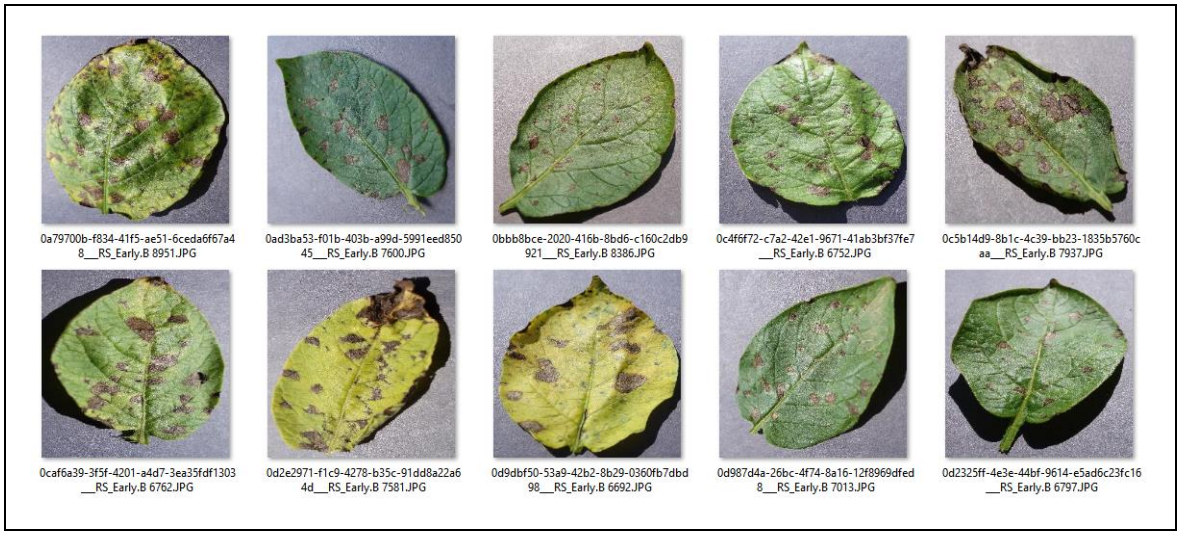
Penelitian dilakukan dengan metode kuantitatif tipe eksperimental menggunakan dataset sekunder. Berikut tahapan pada penelitian diilustrasikan pada Gambar 3.1



Gambar 3.1 Tahapan penelitian

3.1.1 Pengumpulan Data

Proses pengumpulan data pada penelitian ini menggunakan dataset sekunder yang bersumber dari website Kaggle dengan nama Plant Village Dataset yang menyediakan sumber data terbuka atau dataset benchmark. Pada website ini dipilih dataset Kentang dengan jumlah klasifikasi 3 class. Data ini berjumlah 2152, dan memiliki memiliki class, yaitu kering daun (*Early Blight*), busuk daun (*Late Blight*),dan sehat (*Healthy*). *Early Blight* memiliki 1000 citra,*Late Blight* memiliki 1000 citra, dan *Healthy* memiliki 152 citra.



Gambar 3.2 Citra Daun Kentang

3.1.2 Data Preprocessing

Sebelum dilakukannya data *preprocessing*, terlebih dahulu dilakukan pembagian data menjadi data *training* dan data *validation*. Penelitian ini menggunakan skenario dataset yakni 80% : 20%.Perbandingan data tersebut didasarkan pada *pareto principle* yang biasa digunakan dalam data mining, dimana prinsip tersebut menyatakan bahwa 80% kejadian dihasilkan dari 20% sisanya. Pembagian dataset pada penelitian ini dijelaskan pada Tabel 3.1 :

Tabel 3.1 Pembagian data latih dan data uji

No	Nama	Jenis	Jumlah
1	Citra Kentang Early Blight	Training	800
2	Citra Kentang Late Blight	Training	800
3	Citra Kentang Sehat	Training	114
4	Citra Kentang Early Blight	Validation	100
5	Citra Kentang Late Blight	Validation	100
6	Citra Kentang Sehat	Validation	22
7	Citra Kentang Pengujian	Testing	152

Setelah dilakukan pembagian data, kemudian dilakukan proses lainnya yaitu :

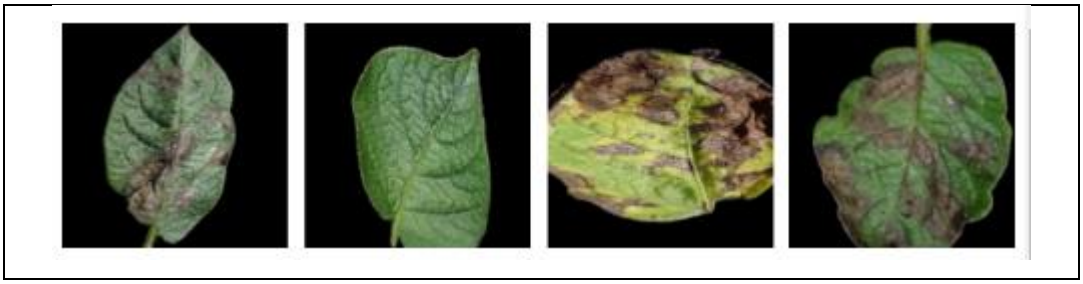
a Augmentasi Data

Proses augmentasi data citra daun untuk proses *training* dilakukan untuk mencegah terjadinya *overfitting* (memiliki kinerja baik selama pelatihan,tetapi buruk pada data baru). Beberapa teknik yang dilakukan pada proses augmentasi data ini dijelaskan pada Tabel 3.2 berikut :

Tabel 3.2 Teknik Pada Proses Data Augmentasi

No	Parameter	Nilai	Keterangan
1	Rescale	1./255	Berfungsi untuk mengubah ukuran data piksel RGB gambar (0-255) menjadi rentang angka (0-1) untuk memudahkan proses training data
2	Rotation	10°	Berfungsi untuk mengubah rotasi gambar kekiri dan kenanan sebesar 10°
3	Width Shift	20%	Berfungsi untuk mengatur posisi gambar pada lebar gambar sebesar 20%
4	Heigth Shift Range	20%	Berfungsi untuk mengatur posisi gambar pada tinggi gambar sebesar 20%
5	Shear Range	10%	Berfungsi meregangkan citra kekanan dan ke kiri sebesar 10%
6	Zoom	10%	Berfungsi untuk memperbesar ukuran citra sebesar 10%

Hasil dari proses tabel diatas dapat dilihat pada Gambar 3.3 berikut :



Gambar 3.3 Citra Daun Kentang yang Sudah di Augmentasi

Pada Gambar 3.3 diatas dapat dilihat perubahan yang terjadi pada daun kentang yang sudah di augmentasi,terlihat gambar yang sudah di *zoom* dan di *flip*.

b Rescale

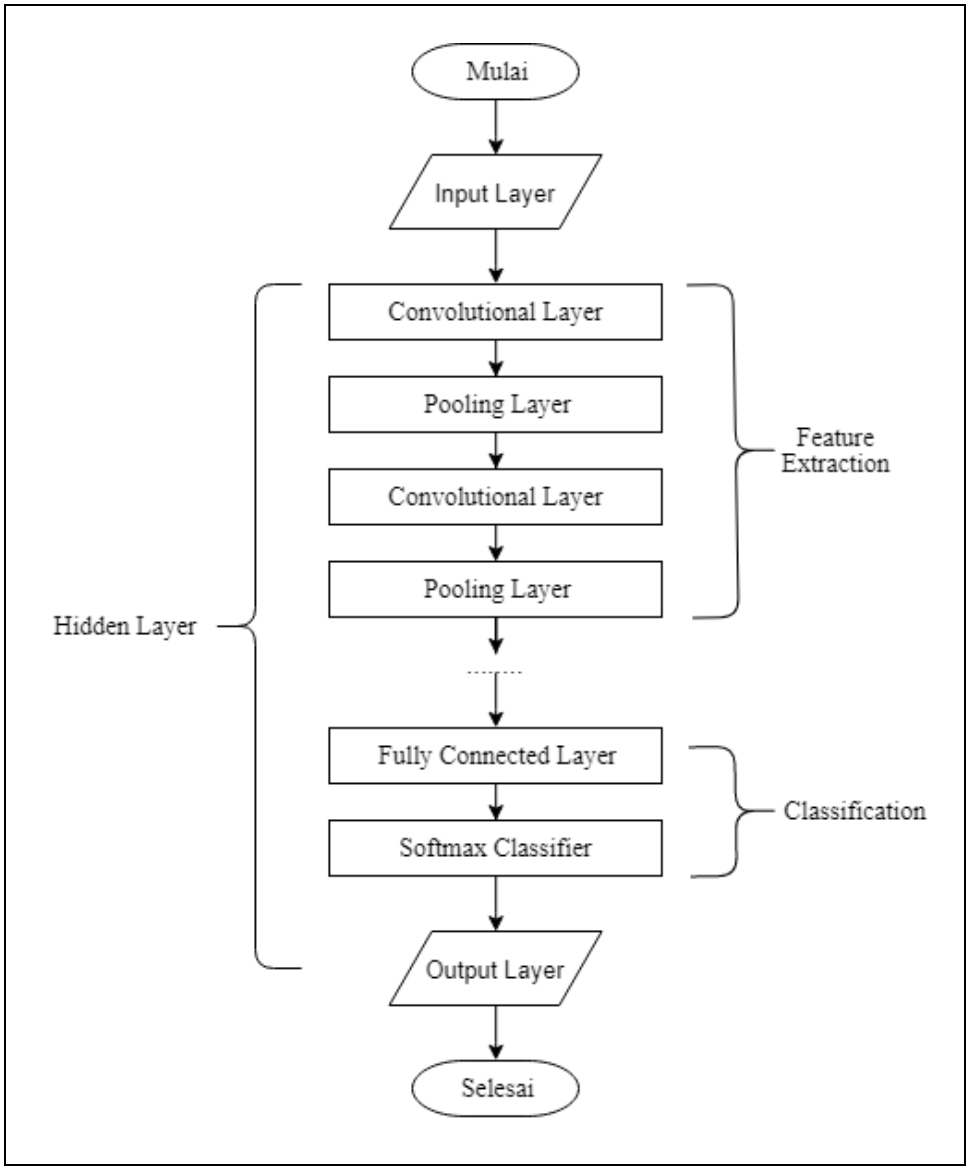
Rescale atau juga biasa disebut *min-max normalization*, proses ini membagi setiap pixel yang memiliki rentang nilai 0 hingga 255 dengan nilai maksimumnya menjadi 0 sampai 1, hal tersebut bertujuan supaya model dapat melakukan proses training menjadi lebih cepat. Proses yang terjadi pada rescale dijelaskan pada Tabel 3.3 berikut menggunakan persamaan 2.2 :

Tabel 3.3 Proses Rescale

Citra Awal				Proses				Citra Hasil		
210	110	115		210-0/255-0	110-0/255-0	115-0/255-0		0,82	0,43	0,45
90	210	80		90-0/255-0	210-0/255-0	80-0/255-0		0,35	0,82	0,31
100	180	150		100-0/255-0	180-0/255-0	150/255		0,39	0,71	0,59

3.1.3 Pembangunan Model CNN

Proses pembangunan model merupakan proses melakukan pembelajaran mesin untuk menemukan pola tertentu didalam data. Tahapan yang dilakukan dalam proses pembangunan model yang pertama adalah menggunakan data citra yang telah melalui tahap *preprocessing* data untuk dilakukan proses *training*, yang mengimplementasikan metode CNN. Setelah proses *training* dilakukan, maka akan menghasilkan model,proses *training* pada model CNN memiliki beberapa tahapan, berikut adalah flowchart dari proses *training* model CNN pada gambar 3.4 berikut :



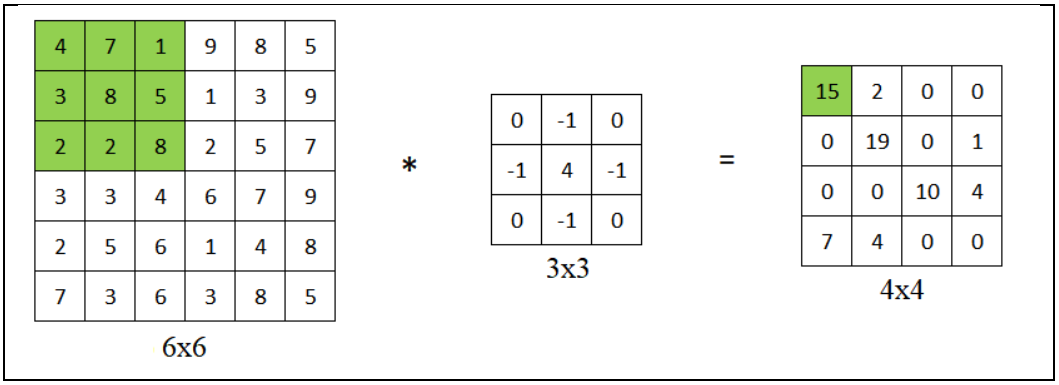
Gambar 3.4 *Flowchart* dari CNN

a. Input Image

Tahapan pertama dalam melakukan pelatihan terhadap model adalah memasukkan data citra daun ke input *layer*,pada *layer* ini citra gambar dikonversi kedalam matriks tiga dimensi,dengan ukuran panjang x lebar x 3 *channels* RGB (*Red,Green,Blue*).Pada penelitian ini nilai RGB pada setiap piksel dinormalisasi menjadi 0-1 untuk mempermudah proses komputasi yang dijelaskan pada Tabel 3.3.

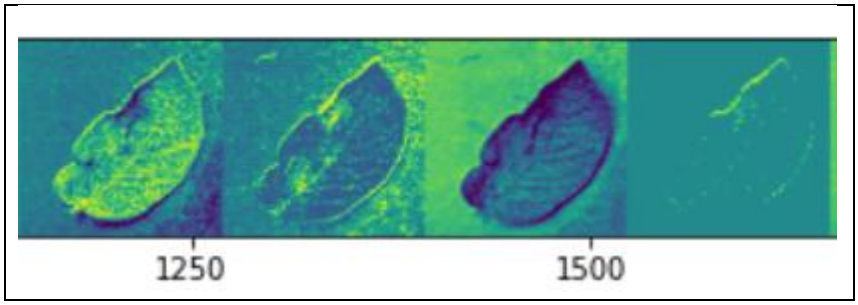
b. Convolution

Proses konvolusi berfungsi untuk mengektrasi fitur-fitur (*feature map*) yang ada pada citra dengan menggunakan *filter*.Penelitian ini menggunakan *filter size* berukuran 3x3 *pixel*,dengan *stride* 1 dan tanpa padding.Pada CNN sendiri *filter size* juga disebut sebagai kernel atau *weight* yang nilainya diinisialisasi secara acak. Contoh prosesnya dapat dilihat pada Gambar 3.5 berikut.



Gambar 3.5 Contoh Proses Konvolusi

Pada ilustrasi gambar 3.5 terdapat input citra berukutan 6x6 yang akan di konvolusi menggunakan *filter size* berukuran 3x3 yang akan menghasilkan *feature map* berukuran 4x4. Operasi konvolusi dilakukan dengan menggeser kernel konvolusi pixel per pixel. Hasil konvolusi disimpan di dalam matriks yang baru. Proses perhitungannya adalah sebagai berikut (0x4)+(-1x7)+(0x1) + (-1x3)+(4x8)+(-1x5) + (0x2)+(-1x2)+(0x8) = 15.Dari proses tersebut menghasilkan nilai 15 yang akan menempati 1 buah sel di citra yang baru.Setelah itu filter digeser lagi kekanan dan kebawah hingga terbentuk 1 citra baru.Berdasarkan ilustrasi diatas dapat kita lihat hasil konvolusi daun kentang pada gambar 3.6



Gambar 3.6 *Feature Map* pada lapisan konvolusi

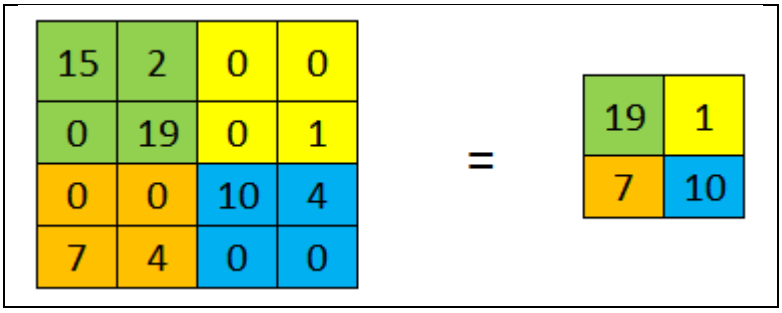
c. Activation function

Untuk mendapatkan objek dalam citra diperlukan pemisahan objek dengan latar belakang pada objek. Pada penelitian kali ini *activation function* ReLU digunakan untuk

menentukan aktif tidaknya neuron pada *neural networks* sehingga hanya neuron yang berhubungan dengan objek saja yang dipilih

d. Pooling Layer

Pooling layer berfungsi untuk mengurangi ukuran spasial dari citra dan mengurangi jumlah parameter dan perhitungan dalam *neural networks*. Pada lapisan ini menerima input dari hasil *feature map* dari hasil konvolusi pada *convolution layer*.Penelitian ini menggunakan *max pooling* ukuran *pooling* 2x2 sehingga citra yang dihasilkan lebih kecil.Sebagai contoh pada Gambar 3.7 merupakan input max pooling,proses ini menghasilkan *output* dengan nilai yang terbesar dari semua nilai input, pada kasus ini 19 adalah nilai terbesar sehingga nilai tersebut lah yang diambil.

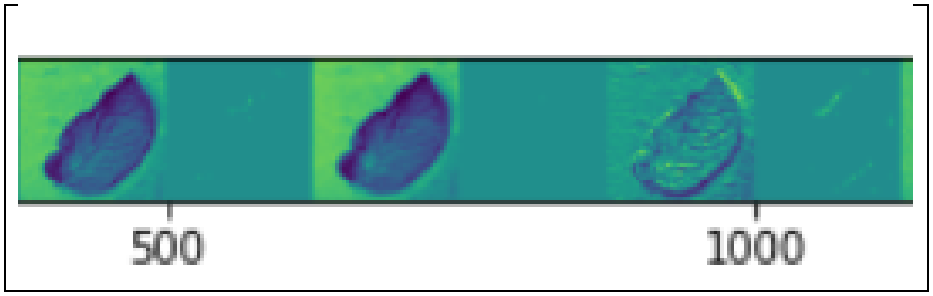


Gambar 3.7 Contoh Proses Pooling

Berdasarkan Gambar 3.7 dengan elemen berwarna hijau nilai terbesarnya adalah 19 maka nilai yang digunakan untuk input citra baru adalah 19 begitupun untuk elemen berwarna kuning dengan nilai terbesar adalah 1 maka input citra barunya adalah 1,begitupun seterusnya.Proses komputasi yang terjadi pada pooling layer dijelaskan seperti pada Tabel 3.4,dan untuk gambar pooling yang sudah dilakukan pada daun kentang dapat dilihat pada gambar 3.5

Tabel 3.4 Proses Pooling Layer

No	Sel	Nilai Area Matriks	Hasil
1	f(1,1)	15,2,0,19	19
2	f(1,2)	0,0,0,1	1
3	f(1,3)	0,0,7,4	7
4	f(1,4)	10,4,0,0	10



Gambar 3.8 Hasil Pooling

e. *Fully Connected Layer*

Tahapan selanjutnya adalah *fully connected layer*,tahapan awal adalah mengubah data matriks 3 dimensi pada tahap konvolusi menjadi satu dimensi vektor (*flatten*).Komputasi *fully connected layer* diilustrasikan seperti Tabel 3.5 menggunakan persamaan 2.5.

Tabel 3.5 Proses *Flattening*

Data Matriks				**Data Vektor (x)**								
6	4	9		6	4	9	7	3	5	1	2	7
7	3	5										
1	2	7										

Data vektor tersebut akan melakukan proses perhitungan seperti pada persamaan 3.1, apabila dihitung secara manual akan terlihat seperti berikut :

y = x1w1 + x2w2 +x3w3 +x4w4 +x5w5 +x6w6 +x7w7 +x8w8 +x9w9

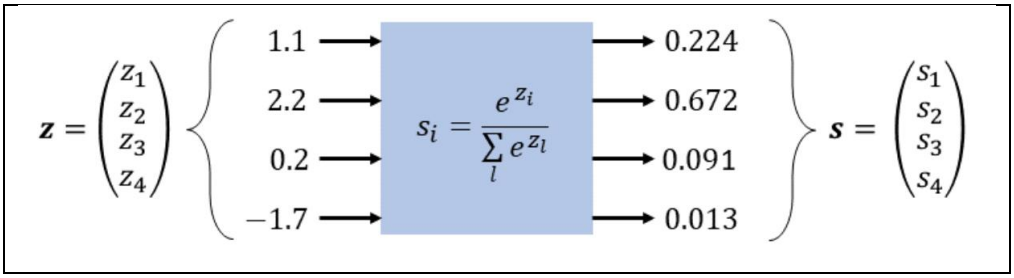
y = (6 . 0,5) + (4 . 0,71) + (9 . 0,25) + (7 . 0,79) + (3 . 6,58) + (5 . -1,25) + (1 . 3,11) + (2 . -2,5) + (7 . 5,20)

y = 3+2,84+2,25+5,53+19,74+(-6,25)+3,11+(-5)+36,4

y = 61,72

f. Output Layer

Output Layer merupakan lapisan yang merepresentasikan hasil akhir dari proses klasifikasi. Tujuan dari layer ini adalah untuk memprediksi output klasifikasi dari segi nilai probabilitas, dimana nilai probabilitas kelas terbesar merupakan output prediksi kelas yang diperoleh. Penelitian ini menggunakan fungsi aktivasi Softmax,proses yang terjadi diilustrasikan seperti Gambar 3.9 berikut :



Gambar 3.9 Proses Klasifikasi Menggunakan Softmax

Keterangan :

Z = Layer Output

S = Probabilitas

3.1.4 Membuat Rencana Kombinasi Pengujian Nilai Epoch,Jenis Optimizer dan Layer Konvolusi

Pada tahap ini, dilakukan perbandingan beberapa parameter pada arsitektur CNN agar didapatkan model yang terbaik. Adapun parameter-parameter yang dilakukan

perbandingan adalah nilai *epoch*, jenis *optimizer* dan skenario *Convolutional Layer*.Kombinasi yang akan diujikan pada penelitian ini dapat dilihat pada Tabel 3.6 berikut ini.

Tabel 3.6 Kombinasi Pengujian

No	Hyperparameter	Parameter 1	Parameter 2	Parameter 3
1	*Optimizer*	Adam	RMSProp	SGD
2	*Epoch*	25	50	100
3	*Convolutional Layer*	4	4	5

3.1.5 Rencana Pengujian dan Indikator Keberhasilan Penelitian

Parameter pertama yang akan dilakukan perbandingan untuk memperoleh arsitektur terbaik adalah parameter *Optimizer*.Pada penelitian ini jenis *Optimizer* yang akan dilakukan perbandingan adalah Adam,RMSProp dan SGD.Hasil training CNN menunjukkan bahwa pemilihan jenis optimizer akan memberikan tingkat akurasi dan losses yang berbeda-beda (Wikarta et al., 2020).Parameter selanjutnya yang akan dilakukan perbandingan adalah *epoch* karena penguunaan epoch sangat bepengaruh terhadap akurasi yang dihasilkan (Wasil et al., 2022).Parameter *epoch* Pada penelitian ini menggunakan perbandingan sebesar 25,50 dan 100.Parameter terakhir yang akan dilakukan perbandingan adalah *layer* konvolusi dengan perbandingan *layer* sebanyak 4 dan 5.Berdasarkan parameter yang diujikan pada tabel 3.6 menghasilkan kombinasi sebanyak 18 kombinasi yang dapat dilihat pada tabel 3.7 berikut :

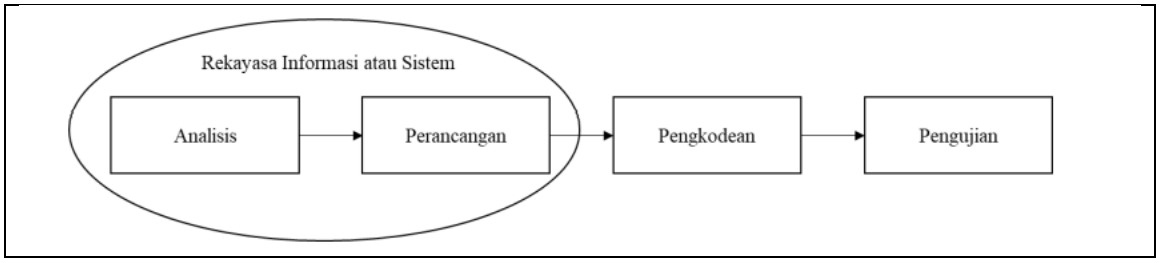
Tabel 3.7 Rencana Pengujian Kombinasi *Hyperparameter* dan *Optimizer*

No	*Optimizer*	*Epoch*	*Convolutional Layer*	Akurasi
1	Adam	25	4	93,69 %
2	Adam	25	5	
3	Adam	50	4	
4	Adam	50	5	
5	Adam	100	4	
6	Adam	100	5	
7	RMSprop	25	4	
8	RMSprop	25	5	
9	RMSprop	50	4	
10	RMSprop	50	5	
11	RMSprop	100	4	
12	RMSprop	100	5	
13	SGD	25	4	
14	SGD	25	5	
15	SGD	50	4	
16	SGD	50	5	
17	SGD	100	4	
18	SGD	100	5	

Dari Hasil training tersebut akan didapatkan akurasi dari masing-masing kombinasi yang diujikan.Kombinasi dengan tingkat akurasi tertinggi dinyatakan sebagai model terbaik,sehingga penelitian ini menjadikan tingkat akurasi pengujian sebagai indikator keberhasilan penelitian

3.2 Pengembangan Perangkat Lunak

Pengembangan perangkat lunak dilakukan dengan menggunakan metode *Linear Sequential Model*, yang memiliki proses sistematis dengan pendekatan sekuensial. Dimulai dari tahap analisis, perancangan, implementasi atau pengkodean, dan pengujian (Widiyanto, 2018).



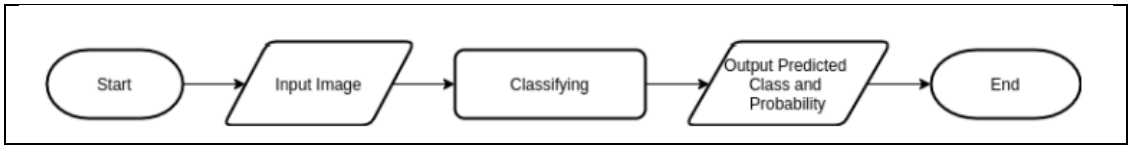
Gambar 3.10 Metodologi Pengembangan Sistem Linear Sequential Model

3.2.1 Analisis

Tahap ini mengumpulkan dan mendefinisikan sistem dengan analisis kebutuhan sistem yang mendalam sehingga sistem yang dibangun sesuai dengan kebutuhan. Tahap ini akan menghasilkan spesifikasi kebutuhan berdasarkan information domain, fungsi, perilaku, performance, dan antarmuka sistem.

3.2.2 Perancangan

Tahap perancangan menggambarkan representasi sistem berdasarkan spesifikasi kebutuhan yang telah dikumpulkan. Representasi tersebut dapat berupa diagram, pseudocode, atau mockup yang menjadi acuan ketika tahap pengkodean.



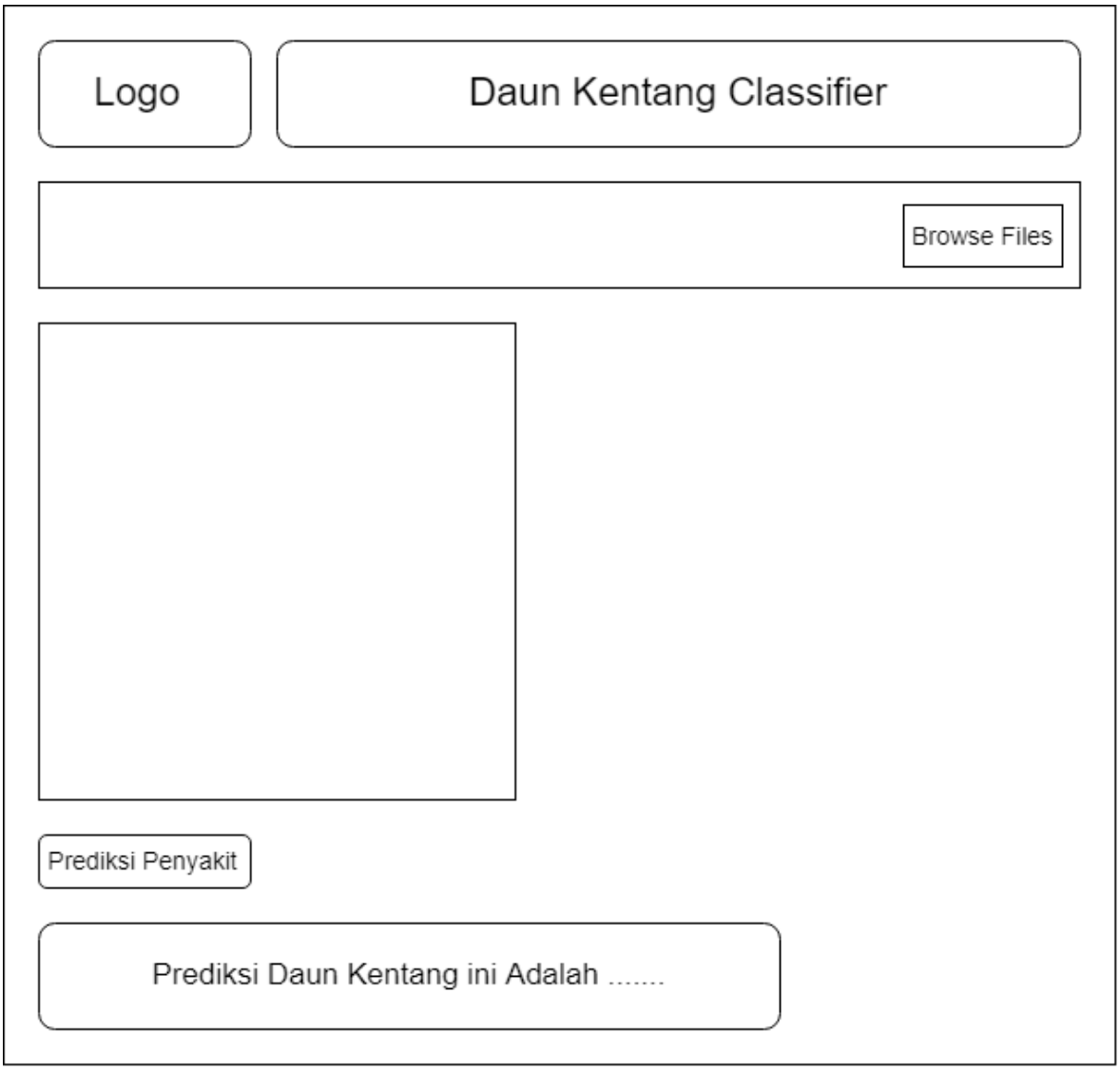
Gambar 3.11 Flowchart Sistem Aplikasi

3.2.3 Implementasi

Tahap implementasi berupa tahap pengkodean atau membuat kode program hasil terjemahan perancangan ke dalam bahasa mesin. Dalam proses implementasi ini, akan dibuat fungsi dan prosedur yang sesuai dengan kebutuhan sistem beserta implementasi antarmuka.

a. Rancangan Antarmuka Aplikasi

Berdasarkan hasil analisis kebutuhan sistem,diartikan rancangan antarmuka aplikasi yang ditampilkan pada gambar 3.12 berikut :



Gambar 3.12 Rancangan antarmuka Aplikasi

Pada halaman utama ini user dapat melakukan penginputan citra dengan menekan tombol buka gambar,kemudian pilih citra yang diinginkan untuk diklasifikasi.Setelah gambar daun kentang ditampilkan user dapat melakukan pemeriksaan apakah daun kentang yang telah dipilih terinfeksi penyakit atau normal dengan menekan tombol prediksi penyakit sehingga ditampilkan hasil klasifikasi seperti pada Gambar 3.9.Aplikasi hanya bisa memproses citra yang berekstensi .jpg & .png saja.

3.2.4 Uji Coba

Setelah perangkat lunak dibangun akan dilakukan pengujian. Pengujian berfokus pada *logical internal* dan *functional external*, yaitu dengan melakukan pengujian pada semua fungsi untuk menemukan *error* dan memastikan masukan yang telah ditetapkan memberikan

keluaran yang diharapkan.Pengujian sistem dilakukan dengan black box metode *boundary value analysis*. Metode ini menitik beratkan pada proses input dengan menguji batas atas dan batas bawah dibawah (Jaya,2018).

Tabel 3.8 Rencana Pengujian

Skenario ke	Test Case	Ekspektasi Hasil
1	Input Citra Normal.jpg	Sukses
2	Input Citra Berpenyakit.jpg	Sukses
3	Input Citra png.png	Sukses

Tabel 3.8 merupakan skenario dari pengujian yang akan dilakukan oleh sistem. Apabila citra yang di inputkan bertipe jpg dan png, maka sistem akan mampu melakukan proses klasifikasi dengan benar, apabila sistem diberi input citra bertipe selain itu, maka citra tidak akan bisa di klasifikasi.

Tabel 3.9 Rencana Pengujian Augmentasi

No	Gambar	Jenis Penyakit	Klasifikasi	Kesimpulan
1				
2				
3				
4				
5				

Tabel 3.9 adalah rencana pengujian untuk citra gambar yang sudah di augmentasi menggunakan rotasi dengan besar 30° dan 60°,*brightness* ditambah 40% dan dikurangi 40%,*shear range* 15% dan 30%.Apabila menginputkan citra yang telah di augmentasi, maka sistem akan mampu melakukan proses klasifikasi. Hasil proses klasifikasi akan disimpulkan dalam bentuk akurasi

BAB IV HASIL DAN PEMBAHASAN

4.1 Implementasi CNN

Implementasi CNN dibagi menjadi 2 bagian,yaitu implementasi dari proses *training* model dan implementasi klasifikasi pada perangkat lunak berdasarkan skenario yang telah dirancang pada Bab 3.Dari hasil pengujian yang telah didapatkan selanjutnya diberikan pembahasan dan analisis pengujian untuk mendapatkan hasil penelitian,sehingga mendapatkan kesimpulan yang diberikan pada pembahasan selanjutnya.

4.1.1 Binding Data

Binding data bertujuan untuk mengikat sumber data yang disimpan pada cloud storage ke IDE yang digunakan,dalam hal ini adalah *google colabolatory* dan *google drive cloud storage* sehingga dapat digunakan tanpa proses *uploading*.Proses tersebut dapat dilihat pada algoritma 4.1 berikut :

Algoritma 4.1 Binding Data ke Clound Storage

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

4.1.2 Inisialisasi Data

Inisialisasi Data berfungsi untuk menyimpan suatu data yang akan digunakan untuk proses komputasi kedalam sebuah variabel sehingga dapat dilakukan proses berikutnya.Proses tersebut dapat dilihat pada Algoritma 4.2 berikut ini :

Algoritma 4.2 Proses Inisialisasi Data kedalam Variabel

```
1 base_dir = '/content/drive/MyDrive/Colab Notebooks
              /Daun Kentang/'
2
3 test_dir = os.path.join(base_dir,'Testing')
4 train_dir = os.path.join(base_dir,'Training')
5 val_dir = os.path.join(base_dir,'Validation')
```

4.1.3 Proses Preprocessing dan Augmentasi Data

Salah satu proses utama yang ada pada aplikasi ini adalah preprocessing.Proses ini bertujuan untuk mempersiapkan data sebelum diolah lebih lanjut dan di augmentasi.Augmentasi data berfungsi agar kebutuhan program terhadap data yang besar dapat terpenuhi.Proses tersebut dapat dilihat pada Algoritma 4.3 berikut :

Algoritma 4.3 Preprocessing dan Augmentasi Data

```
1
2 train_datagen = ImageDataGenerator(
3                   rescale = 1./255,
4                   rotation_range=15,
5                   shear_range=0.2,
```

```
6                   zoom_range = 0.1
7  )
8 val_datagen = ImageDataGenerator (
9                 rescale=1./255
10 )
11
12 training_dataset = train_datagen.flow_from_directory(train_dir,
13                                     target_size=(256,256),
14                                     batch_size=64,
15                                     class_mode='sparse')
16 validation_dataset = val_datagen.flow_from_directory(val_dir,
17                                     target_size=(256,256),
18                                     batch_size=64,
19                                     class_mode='sparse')
20
```

Pada algoritma 4.3 dapat kita lihat dilakukan proses augmentasi data citra dengan melakukan rotasi sebesar 15°,*shear range* 20%,*zoom range* 10%,dan normalisasi citra.Setiap kali model akan melakukan *training* citra,maka akan diambil 64 citra yg berbeda dengan data training di awal sesuai dengan besar *batch_size.*

4.1.4 Inisialisasi Model CNN

Inisialisasi model CNN diperlukan sebelum memasuki proses *training.*Proses pembuatan model CNN ini diawali dengan *preprocessing* yang telah dibahas pada sub bab 4.1.3.Dimana terdiri dari *rescale,rotation,shear,dan zoom* kemudian pada proses ini dilakukan input *hyperparameter* yang akan diujikan dari *number of filters,filter size,*inisialisasi *pooling layer* dan sebagainya.Proses tersebut dapat kita lihat pada Algoritma 4.4 berikut :

Algoritma 4.4 Inisialisasi Model CNN

```
model = tf.keras.models.Sequential([

        # 4 Layer
        tf.keras.layers.Conv2D(16,(3,3),activation = 'relu',input_shape = (256,
256,3)),
        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(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.Flatten(),
```

```
        tf.keras.layers.Dense(64,activation='relu'),
        tf.keras.layers.Dropout(0.1),
        tf.keras.layers.Dense(3,activation='softmax')
])

#print model summary
model.summary()

model.compile(loss = 'sparse_categorical_crossentropy',
               optimizer = 'Adam',
               metrics = ['accuracy'])
```

Berdasarkan algoritma 4.4, setelah melakukan input *hyperparameter* kemudian dilakukan proses *compile* menggunakan Adam *Optimizer*.

4.1.5 Proses Training Model CNN

Setelah dilakukannya proses inisialisasi model CNN, model tersebut dapat melalui tahap selanjutnya yaitu proses training. Proses ini dilakukan dengan menginputkan beberapa nilai *hyperparameter* lainnya yaitu number of *epochs* dan *number of validation steps*. Proses tersebut disajikan pada Algoritma 4.5 berikut :

Algoritma 4.5 Inisialisasi Proses Training Model CNN

```
# Untuk save model yang telah dibuat

model_version = "Kombinasi5"
model_versionSave = callbacks.ModelCheckpoint(f"/content/drive/MyDrive/Colab No
tebooks/ModelsCNN/{model_version}.hdf5", save_best_only=True, monitor='val_accu
racy', mode='max')

history = model.fit(
            training_dataset,
            steps_per_epoch = 15, #20
            epochs = 100, #70
            validation_data = validation_dataset,
            callbacks=[model_versionSave])
```

Berdasarkan algoritma 4.5 program melakukan *training* pada model CNN berdasarkan data yang diinputkan sehingga menghasilkan empat nilai berikut pada *console colaboratory* seperti pada Tabel 4.1 berikut :

Tabel 4.1 Hasil Training Model CNN

No	Loss	Accuracy	Validation Loss	Validation Accuracy
1	0.9288	0.5219	0.8927	0.6937
2	0.7165	0.7357	0.6825	0.7297
3	0.5470	0.8002	0.6185	0.7342
…	…	…	…	…
84	**0.0073**	**0.9979**	**0.0624**	**0.9820**
…	…	…	…	…
98	0.0138	0.9979	0.2868	0.9414
99	0.0095	0.9958	0.4837	0.9189
100	0.0035	1.0000	0.4426	0.9324

Berdasarkan Tabel 4.1 dari 100 *epochs* yang dilakukan proses *training* terlihat bahwa *epoch* 84 mendapatkan akurasi tertinggi sebesar 98,20% dengan *loss* sebesar 7%

4.1.6 Evaluasi Model CNN

Tahapan evaluasi model dilakukan dengan mengeksekusi kode untuk menghasilkan *confusion matrix*,tingkat akurasi. Ketiga hal ini menjadi poin penting pada proses evaluasi model CNN. Sebelum melakukan evaluasi, Model yang sudah disimpan sebelumnya dengan menggunakan ModelCheckpoint akan dipanggil terlebih dahulu. Proses pemanggilan tersebut disajikan pada Algoritma 4.6 berikut :

Algoritma 4.6 Evaluasi Akurasi Model CNN

```
#load Model
from keras.models import load_model
modeloptimal = models.load_model(f"/content/drive/MyDrive/Colab Notebooks/Model
sCNN/{model_version}.hdf5")
model=modeloptimal
```

Sehingga dari Algoritma 4.6 tersebut,didapatkan hasil output seperti pada gambar 4.1 Berikut :

```
4/4 [==============================] - 4s 810ms/step - loss: 0.0624 - accuracy: 0.9820
Test Accuracy: 98.2%
```

Gambar 4.1 Output Evaluasi Model CNN

Dari Algoritma 4.6 tersebut,menghasilkan output seperti pada gambar 4.1,sehingga diketahui hasil bahwa tingkat akurasi model paling optimal yang dihasilkan pada penelitian ini adalah 98,20 % dengan nilai loss 6,24 %.Setelah diketahui tingkat akurasi yang dihasilkan oleh model,hasil prediksi dari keseluruhan dataset yang digunakan sebagai data uji dapat dilihat dengan mengeksekusi program seperti pada Algoritma 4.7 berikut :

Algoritma 4.7 Proses Visualisasi Confusion Matrix

```
# Confusion Matrix

from sklearn.metrics import confusion_matrix,classification_report
import seaborn as sns

prediksi = model.predict_generator(testing_dataset)
y_pred = np.argmax(prediksi,axis=1)


print("======================== Confusion Matrix ========================")
print(confusion_matrix(testing_dataset.classes,y_pred))
```

Evaluasi dengan *Confusion Matrix* dilakukan dengan data uji 222 citra daun,untuk setiap kelas citra daun berpenyakit 100 citra dan citra daun sehat 22.

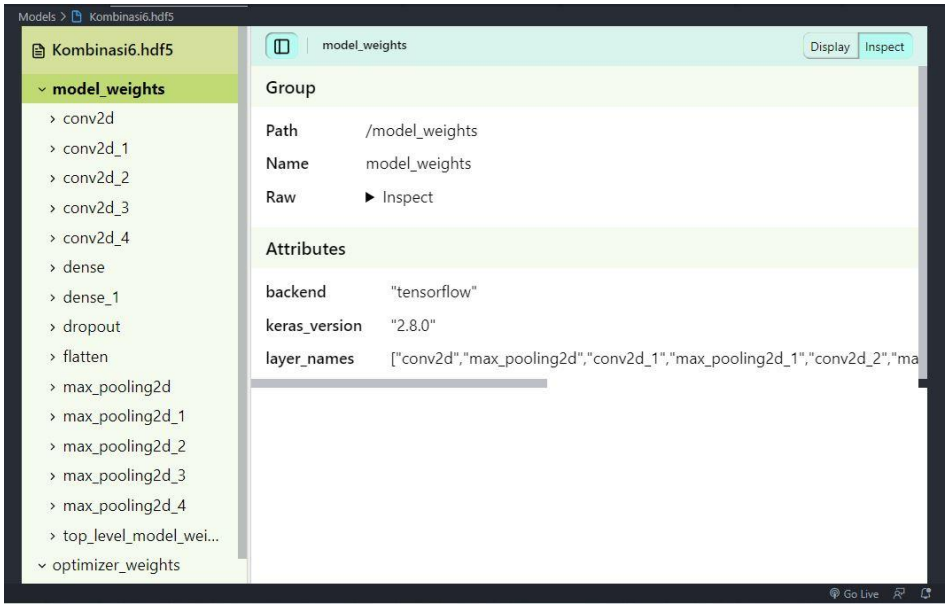
4.2 Implementasi Perangkat Lunak

Bagian ini menjelaskan hasil dari pengembangan sistem yang telah dirancang sebelumnya. Sistem yang dikembangkan digunakan untuk pengujian model dan menampilkan hasil klasifikasi pada penelitian ini. Implementasi pada perangkat lunak dapat dilakukan setelah model CNN selesai dilakukan proses *training*,hal ini dimungkinkan apabila pengguna menyimpan hasil proses *training* tersebut kedalam sebuah *file* dengan *format* .hdf5. Pada penelitian ini menggunakan fungsi *Callbacks* untuk melakukan penyimpanan *File* yang dapat dilihat pada Algoritma 4.8 berikut :

Algoritma 4.8 Penyimpanan Model CNN

```
# Untuk save model yang telah dibuat
model_version = "Kombinasi6"
model_versionSave = callbacks.ModelCheckpoint(f"/content/drive/MyDrive/Colab No
tebooks/ModelsCNN/{model_version}.hdf5", save_best_only=True, monitor='val_accu
racy', mode='max')
```

Model dengan format .hdf5 yang sudah disimpan dapat kita lihat pada gambar 4.2 berikut :



Gambar 4.2 Model CNN

Pada Gambar 4.2 dapat kita lihat bahwa model menyimpan arsitektur yang telah kita *training*.Pada model kombinasi6.hdf5 terdapat 5 kali konvolusi dan 5 kali *max-pooling*.Terlihat juga *optimizer* yang digunakan pada arsitektur ini adalah *optimizer* adam.*File* dengan format .hdf5 tersebut diimplementasikan pada perangkat lunak dengan menuliskan kode seperti pada Algoritma 4.9 berikut

Algoritma 4.9 Proses Load Model .hdf5 Pada Perangkat Lunak

```
label = teachable_machine_classification(
            image, 'Models/Kombinasi6.hdf5')
```

Path *File* pada citra yang akan diklasifikasikan tersebut disimpan lalu dilakukan proses klasifikasi pada *function predict*. Proses klasifikasi tersebut dilakukan dengan mengeksekusi kode pada Algoritma 4.10 berikut :

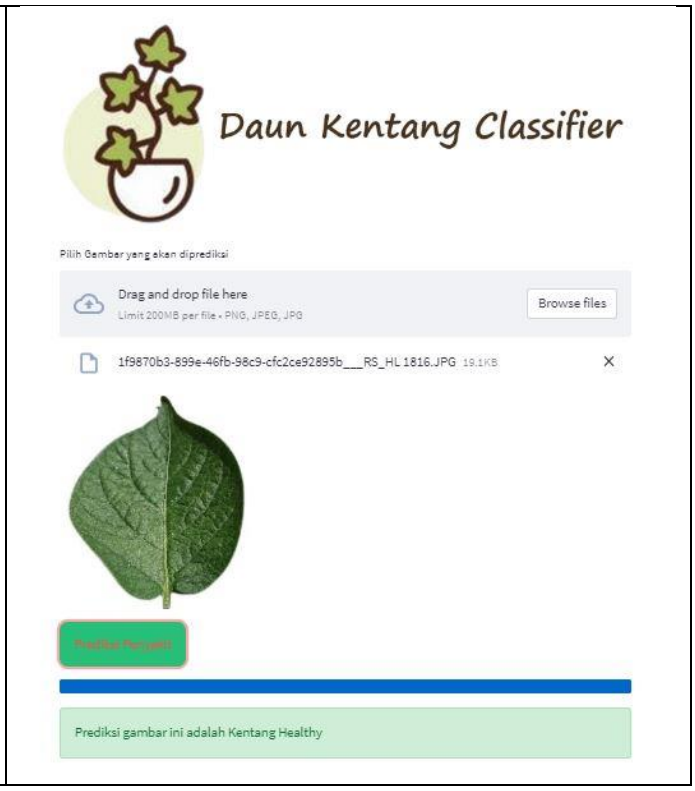
Algoritma 4.10 Proses Klasifikasi dan Menampilkan Hasil

```
def teachable_machine_classification(img, weights_file):

    model = keras.models.load_model(weights_file)
    image = img
    size = (256, 256)
    image = ImageOps.fit(image, size, Image.ANTIALIAS)
    image = np.asarray(image)
    image = np.expand_dims(image, axis=0)
    image = image/255
    prediction = model.predict(image)
    return np.argmax(prediction)

if label == 0:
```

```
        st.warning("Prediksi gambar ini adalah Kentang Early Blight")
    if label == 1:
        st.success("Prediksi gambar ini adalah Kentang Healthy")
    if label == 2:
        st.warning("Prediksi gambar ini adalah Kentang Late Blight")
```

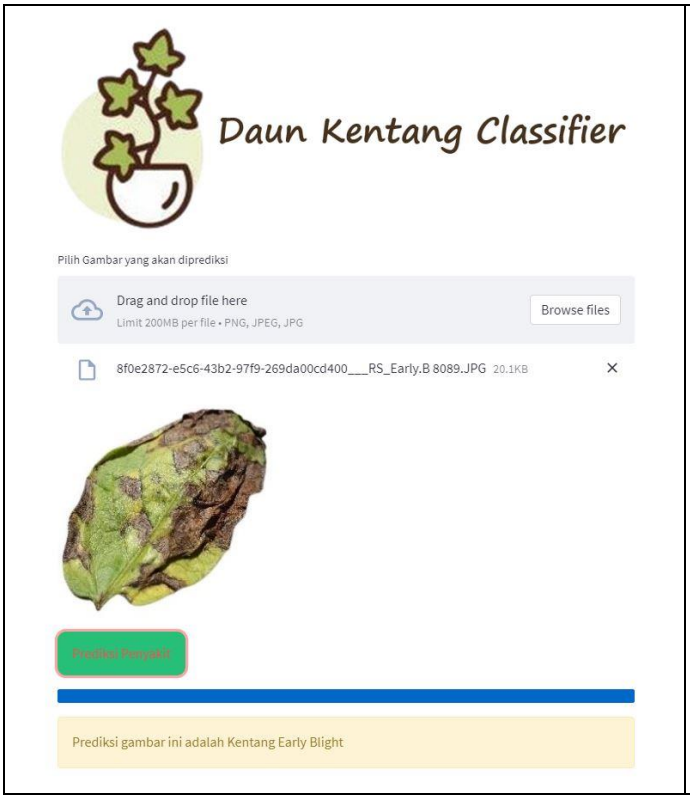
Sehingga apabila program aplikasi dijalankan dan dilakukan proses klasifikasi, akan terlihat pada Gambar 4.3,Gambar 4.4,Gambar 4.5,dan Gambar 4.6 berikut :

Gambar 4.3 Halaman Awal Untuk Input Data Citra

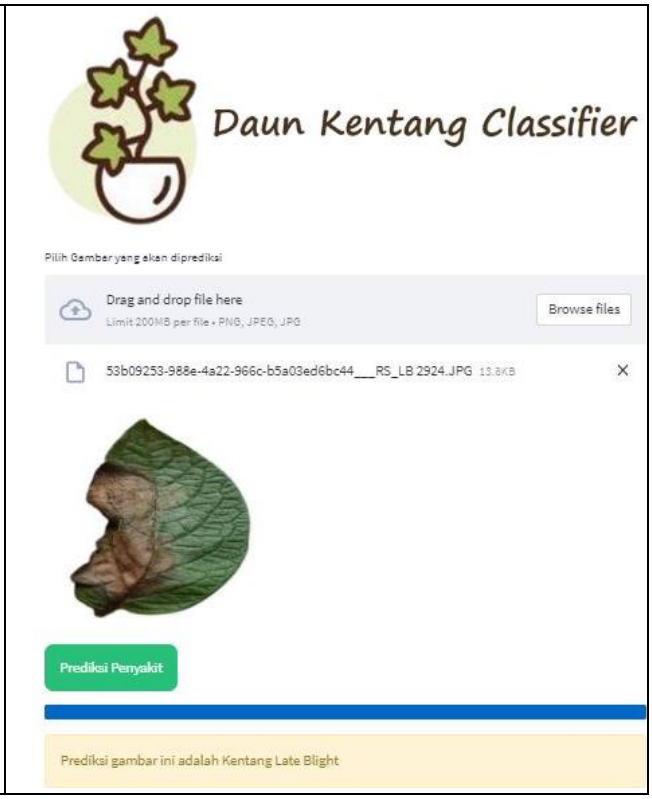


Gambar 4.4 Hasil Proses Klasifikasi Kentang Healthy

Pada Gambar 4.3 Merupakan halaman awal untuk input data citra yang akan diklasifikasikan.Pada Gambar 4.3 dapat dilihat menu untuk browse files yang akan digunakan untuk input gambar kemudian setelah gambar yang ingin diklasifikasi diinput,user harus menekan tombol "Prediksi penyakit" sehingga halaman akan berubah seperti pada Gambar 4.4 dan sistem akan menampilkan hasil klasifikasi dibawah.Pada Gambar 4.4 hasil klasifikasi citra daun kentang tersebut adalah daun kentang Healthy.Pada Gambar 4.5 adalah hasil klasifikasi daun kentang dengan penyakit *Early Blight*,begitu juga dengan Gambar 4.6 sistem mengklasifikasi citra daun kentang dengan penyakit *Late Blight*.Berbeda dengan daun citra tanpa penyakit/sehat,pada Gambar 4.5 dan Gambar 4.6 tulisan dan backgroundnya berwarna kuning.



Gambar 4.5 Hasil Proses Klasifikasi Kentang Early Blight



Gambar 4.6 Hasil Proses Klasifikasi Kentang Late Blight

4.3 Hasil Pengujian Model CNN

Pada skenario ini adalah membandingkan nilai akurasi pada saat menggunakan *Optimizer* Adam,RMSprop,dan SGD dengan *Epoch* 25,50,dan 100 dengan *Convolution Layer* menggunakan 5 layer.Sehingga total pengujian yang dilakukan pada penelitian ini sebanyak 18 kali.Tujuan dari penentuan parameter model ini ingin membandingkan model mana yang memiliki akurasi terbaik dengan memperhatikan nilai parameternya.Serta mengetahui apakah dengan dilakukannya penambahan *epoch*,mengganti *optimizer* serta mengubah jumlah layer konvolusi dapat meningkatkan kualitas model CNN dalam mengklasifikasi citra daun kentang.Keseluruhan hasil dapat dilihat pada Tabel 4.2 berikut :

Tabel 4.2 Hasil Pengujian Model CNN

No	*Optimizer*	*Epoch*	*Convolutional Layer*	Akurasi
1	Adam	25	4	93,69 %
2	Adam	25	5	91,44 %
3	Adam	50	4	94,14 %
4	Adam	50	5	92,79 %
5	Adam	100	4	96,85 %
6	**Adam**	**100**	**5**	**98,20 %**
7	RMSprop	25	4	85,59 %
8	RMSprop	25	5	88,74 %
9	RMSprop	50	4	90,99 %
10	RMSprop	50	5	86,94 %
11	RMSprop	100	4	94,40 %

12	RMSprop	100	5	94,59 %
13	SGD	25	4	83,33 %
14	SGD	25	5	81,08 %
15	SGD	50	4	83,78 %
16	SGD	50	5	82,43 %
17	SGD	100	4	89,19 %
18	SGD	100	5	83,78 %

Dari 18 percobaan yang telah dilakukan,didapatkan kombinasi terbaik yaitu pada pengujian ke 6 dengan menggunakan *Optimizer* Adam,*Epoch* 100,dan menggunakan 4 dan 5 layer konvolusi.Akurasi yang didapatkan sama yaitu 98,2%.Akurasi ini adalah yang paling tinggi,meningkat sekitar 4% dibanding akurasi tinggi lainnya yang berkisar diangka 94%.Akurasi dengan *Optimizer* SGD adalah yang paling rendah dan belum mencapai tingkat akurasi 90%.

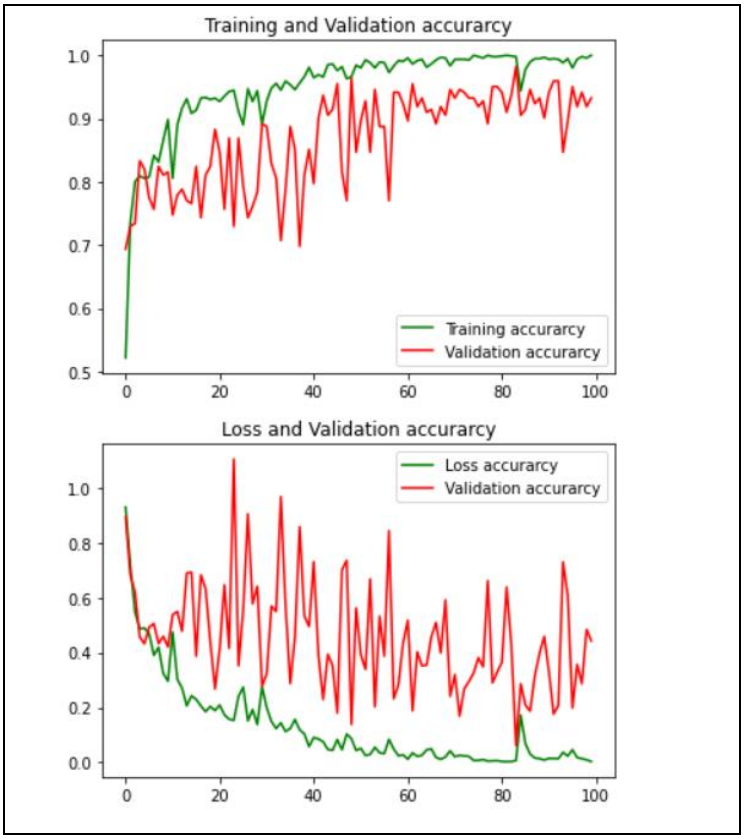
Berdasarkan kombinasi parameter yang telah diujikan pada Tabel 4.2 diatas dapat disimpulkan bahwa semakin menuju nilai 100 *epoch* yang digunakan maka akurasi dari hasil *testing* semakin tinggi.Dapat dilihat pada pengujian ke 1 dengan epoch 25 mendapatkan akurasi 93,69%,pengujian ke 3 dengan epoch 50 mendapatkan akurasi 94,14%,pengujian ke 5 dengan epoch 100 mendapatkan akurasi 96,85%.

Kemudian berdasarkan kombinasi parameter *convolutional layer* yang pada penelitian ini menggunakan 2 layer konvolusi didapatkan kesimpulan bahwa penggunaan 4 layer konvolusi menghasilkan akurasi lebih baik daripada menggunakan 5 layer konvolusi.Berdasarkan Tabel 4.2 diatas pengujian 1 dengan 4 layer konvolusi mendapatkan akurasi 93,69 % sedangkan pengujian 2 dengan 5 layer konvolusi mendapatkan akurasi 91,44 %.

Selanjutnya kombinasi parameter *Optimizer* dengan 3 jenis *Optimizer* dapat disimpulkan bahwa *Optimizer* Adam menghasilkan akurasi tertinggi.Jika dilihat dari Tabel 4.2 pada pengujian 1 dan pengujian 2 mendapatkan akurasi diatas 90% sedangkan untuk *Optimizer* lain pada pengujian 7,8,13,dan 14 mendapatkan akurasi dibawah 90%.Nilai tersebut menunjukkan bahwa *Optimizer* SGD memiliki akurasi terendah, sementara *Optimizer* adam memiliki akurasi tertinggi.

4.3.1 Analisis Pengujian Model CNN

Analisis pengujian dilakukan untuk menentukan kualitas model CNN yang telah dirancang,proses analisis dilakukan dengan menganalisa kurva yang didapatkan pada saat proses *training* dilakukan,berdasarkan pengujian yang telah dilakukan sebanyak 18 kali pada Tabel 4.2. Didapatkan kombinasi *hyperparameter* dan *optimizer* optimal yaitu pada pengujian ke 6 ,dari pengujian tersebut didapatkan hasil training berupa *accuracy,loss,validation accuracy* dan *validation loss* yang di visualisasikan seperti Gambar 4.7 berikut :



Gambar 4.7 Hasil Visualisasi *Training* Model dan *Loss* Model CNN Pengujian ke 6

Hasil Visualisasi dari Gambar 4.7 dijelaskan sebagai berikut :

a. Diagram x merupakan merupakan nilai atau banyaknya *epochs* yang diberikan pada model
b. Diagram y merupakan nilai *loss* atau akurasi yang dihasilkan dari setiap *epochs* yang berjalan
c. Garis hijau merupakan kurva dari *training* model yang sudah dilakukan.Pada diagram model akurasi,semakin tinggi nilai akurasi,maka semakin baik kualitas model tersebut.Sebaliknya pada diagram model *loss*,semakin rendah nilai *loss* maka semakin baik kualitas model tersebut.
d. Garis merah merupakan kurva dari *validation* model yang dilakukan.Pada diagram tersebut apabila *validation* memiliki selisih nilai yang tidak jauh dengan kurva training,serta perubahan nilai yang stabil,maka model tersebut memiliki kualitas yang baik,tidak *overfitting* maupun *underfitting*.

Model yang baik akan memiliki *loss* rendah dan akurasi yang tinggi (Allaam & Wibowo, 2021).Terlihat pada grafik bahwa model yang dihasilkan cukup baik karena memiliki *loss* rendah dan akurasi yang tinggi yaitu 98,2%.

4.4 Hasil Penelitian

Penelitian ini menyimpulkan bahwa, parameter terbaik yang dapat digunakan pada perancangan model CNN untuk klasifikasi citra daun kentang ditampilkan pada Tabel 4.3 berikut :

Tabel 4.3 Kombinasi Optimal *Hyperparameter* dan *Optimizer*

Optimizer	***Epoch***	***Convolutional Layer***	**Akurasi**
Adam	100	5	98,20 %

Sehingga dari keseluruhan percobaan pertama hingga akhir,kombinasi *hyperparameter* dan *Optimizer* yang memberikan hasil paling optimal adalah kombinasi pada percobaan ke 6 seperti pada Tabel 4.3,untuk mendapatkan nilai akurasi dapat digunakan *confusion matrix* seperti pada Tabel 4.4 berikut :

Tabel 4.4 *Confusion Matrix* Model Optimal

Matriks		Predict Class		
		Early Blight	Healthy	Late Blight
Actual Class	Early Blight	100	0	0
	Healthy	0	22	0
	Late Blight	3	1	96

Berdasarkan Tabel 4.4 diatas hasil klasifikasi dari model menunjukan hasil yang baik. Klasifikasi terhadap citra daun kentang berpenyakit *Early Blight* diklasifikasikan ke dalam *Early Blight* sebanyak 100 data. Klasifikasi pada citra daun kentang tak berpenyakit (*Healthy*) diklasifikasikan sebagai *Healthy* sebanyak 22 data. Kemudian yang terakhir adalah klasifikasi terhadap citra daun kentang berpenyakit *Late Blight* diklasifikasikan benar kedalam *Late Blight* sebanyak 96, dan diklasifikasikan bukan *Late Blight* sebanyak 4 data. Perhitungan akurasi dari keseluruhan matriks diatas apabila dihitung menggunakan persamaan 2.6 adalah sebagai berikut :

$$Akurasi = \frac{Jumlah\ Prediksi\ Benar}{Jumlah\ Prediksi\ Salah} = \frac{218}{222} = 0{,}9820 = 98{,}20\ \%$$

Nilai akurasi yang didapatkan dari hasil perhitungan *confusion matrix* sama dengan hasil nilai tingkat akurasi yang didapatkan dari perhitungan algoritma machine, dimana nilai akurasi yang didapatkan adalah sebesar 0,982 atau 98,20 % sehingga dapat disimpulkan bahwa dengan nilai akurasi yang didapatkan tersebut telah cukup baik dan mampu mengklasifikasikan citra jenis penyakit pada kentang berdasarkan daun tepat sesuai dengan kategorinya

4.4.1 Hasil Pengujian Sistem

Dari hasil pengembangan sistem diatas,selanjutnya dilakukan uji coba *format file* dan uji API *remove background* pada sistem.Hasilnya dapat dilihat pada Tabel 4.5 dan Tabel 4.6 Berikut :

Tabel 4.5 Hasil Pengujian *Format File* pada Sistem

Data Testing	Perkiraan Hasil	Hasil	Kesimpulan
DaunSehat.jpg	True	True	Success
DaunBerpenyakit.jpg	True	True	Success
DaunSehat.png	True	True	Success
DaunBerpenyakit.png	True	True	Success
EkstensiLain.webdl	False	False	Success

Tabel 4.6 Hasil Pengujian API pada Sistem

Data Testing	Perkiraan Hasil	Hasil	Kesimpulan
DaunBG.jpg	True	True	Success
DaunNoBG.png	True	True	Success

Selanjutnya, dilakukan pengujian secara acak dengan berbagai jenis citra daun pada database *testing*.Pengujian dilakukan kepada 6 citra daun dengan 3 citra memiliki background dan 3 citra daun tidak memiliki background.Jenis penyakit daun kentang pada *database* ada tiga antara lain *Early Blight,Healthy,*dan *Late Blight.*Citra daun kentang yang akan dilakukan pengujian dapat dilihat pada tabel berikut :

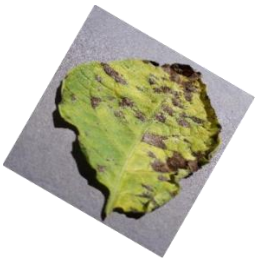
Tabel 4.7 Citra Daun Kentang dalam Pengujian

No	Gambar Daun	Jenis Penyakit	Klasifikasi	Kesimpulan
1		Early Blight	Early Blight	Benar
2		Early Blight	Early Blight	Benar
3		Healthy	Healthy	Benar
4		Healthy	Healthy	Benar

No	Gambar Daun	Jenis Penyakit	Klasifikasi	Kesimpulan
5		Late Blight	Late Blight	Benar
6		Late Blight	Late Blight	Benar

Selanjutnya dataset akan dipilih secara acak kemudian di augmentasi,dataset yg dipilih memasukkan masing-masing 2 gambar untuk setiap penyakit,yang terdiri dari 1 gambar memiliki *background* dan satu lainnya tidak memiliki *background*.Proses augmentasi database *testing* yang dilakukan pada penelitian ini adalah merotasi gambar,melakukan penambahan *brightness*,dan melakukan *shear range*.Setelah proses augmentasi selesai kemudian data di klasifikasi.Peneliti mencatat setiap prediksi yang benar dan prediksi yang salah,kemudian hasil tersebut dikumpulkan dan digunakan untuk menghitung nilai akurasi pada setiap Tabel pengujian.Hasil dari proses klasifikasi dataset yang sudah di augmentasi dapat dilihat pada Tabel 4.8 Berikut :

Tabel 4.8 Hasil Klasifikasi Menggunakan Rotasi 30 Derajat

No	Gambar	Jenis Penyakit	Klasifikasi	Kesimpulan
1		Early Blight	Early Blight	Berhasil
2		Early Blight	Late Blight	Gagal

No	Gambar	Jenis Penyakit	Klasifikasi	Kesimpulan
3		Healthy	Late Blight	Gagal
4		Healthy	Late Blight	Gagal
5		Late Blight	Late Blight	Berhasil
6		Late Blight	Late Blight	Berhasil

Berdasarkan pengujian diatas diperoleh informasi sebagai berikut :
Akurasi : 3/6 * 100% = 50%
Akurasi dari pengujian pada Tabel 4.5 sebesar 50%

Tabel 4.9 Hasil Klasifikasi Menggunakan Rotasi 60 Derajat

No	Gambar	Jenis Penyakit	Klasifikasi	Kesimpulan
1		Early Blight	Early Blight	Berhasil

No	Gambar	Jenis Penyakit	Klasifikasi	Kesimpulan
2		Early Blight	Late Blight	Gagal
3		Healthy	Late Blight	Gagal
4		Healthy	Late Blight	Gagal
5		Late Blight	Late Blight	Berhasil
6		Late Blight	Late Blight	Berhasil

Berdasarkan pengujian diatas diperoleh informasi sebagai berikut :

Akurasi : 3/6 * 100% = 50%

Akurasi dari pengujian pada Tabel 4.6 sebesar 50%

Tabel 4.10 Hasil Klasifikasi Brightness 40%

No	Gambar	Jenis Penyakit	Klasifikasi	Kesimpulan
1		Early Blight	Early Blight	Berhasil
2		Early Blight	Early Blight	Berhasil
3		Healthy	Healthy	Berhasil
4		Healthy	Healthy	Berhasil
5		Late Blight	Late Blight	Berhasil
6		Late Blight	Late Blight	Berhasil

Berdasarkan pengujian diatas diperoleh informasi sebagai berikut :

Akurasi : 6/6 * 100% = 100%

Akurasi dari pengujian pada Tabel 4.7 sebesar 100%

Tabel 4.11 Hasil Klasifikasi Menurunkan Brightness 40%

No	Gambar	Jenis Penyakit	Klasifikasi	Kesimpulan
1		Early Blight	Early Blight	Berhasil
2		Early Blight	Early Blight	Berhasil
3		Healthy	Healthy	Berhasil
4		Healthy	Healthy	Berhasil
5		Late Blight	Late Blight	Berhasil
6		Late Blight	Late Blight	Berhasil

Berdasarkan pengujian diatas diperoleh informasi sebagai berikut :
Akurasi : 6/6 * 100% = 100%
Akurasi dari pengujian pada Tabel 4.7 sebesar 100%

Tabel 4.12 Hasil Klasifikasi *Shear Range* 15 %

No	Gambar	Jenis Penyakit	Klasifikasi	Kesimpulan
1		Early Blight	Early Blight	Berhasil
2		Early Blight	Early Blight	Berhasil
3		Healthy	Healthy	Berhasil
4		Healthy	Late Blight	Gagal
5		Late Blight	Late Blight	Berhasil
6		Late Blight	Late Blight	Berhasil

Berdasarkan pengujian diatas diperoleh informasi sebagai berikut :

Akurasi : 5/6 * 100% = 83,3%

Akurasi dari pengujian pada Tabel 4.9 sebesar 83,3%

Tabel 4.13 Hasil Klasifikasi *Shear Range* 30 %

No	Gambar	Jenis Penyakit	Klasifikasi	Kesimpulan
1		Early Blight	Early Blight	Berhasil
2		Early Blight	Early Blight	Berhasil
3		Healthy	Healthy	Berhasil
4		Healthy	Late Blight	Gagal
5		Late Blight	Late Blight	Berhasil
6		Late Blight	Late Blight	Berhasil

Berdasarkan pengujian diatas diperoleh informasi sebagai berikut :
Akurasi : 5/6 * 100% = 83,3%
Akurasi dari pengujian pada Tabel 4.10 sebesar 83,3%

4.5 Pembahasan

Hyperparameter dan *Optimizer* diperoleh dengan melakukan pengujian sebanyak 18 kali percobaan seperti yang telah dijelaskan pada Tabel 4.2.Ada 3 parameter dimana 2 parameter memiliki 3 nilai dan 1 parameter memiliki 2 nilai sehingga kombinasi yang akan terjadi adalah sebanyak 18 kombinasi.Pengujian ini perlu dilakukan karena kinerja prediktif sangat dipengaruhi oleh nilai *hyperparameter* ataupun *Optimizer* yang digunakan untuk melatihnya. Contohnya *artificial neural network* membutuhkan penentuan jumlah *hidden layer*, *node* dan banyak lagi parameter lain yang terkait dengan proses *fitting* model (Lujan-Moreno et al., 2018)

Berdasarkan hasil dari pengujian kombinasi *hyperparameter* pada setiap model CNN yang dirancang. Setiap *Hyperparameter* dan *Optimizer* yang diujikan memberi pengaruh yang besar pada model yang dihasilkan.Kombinasi paling optimal terjadi pada percobaan ke 6 dengan kombinasi *Optimizer* Adam,4 *Convolution Layer*,dan 100 *epochs* menghasilkan tingkat akurasi tertinggi 98,20%. Akurasi terendah pada percobaan ke 14 dengan kombinasi *Optimizer* SGD,5 *Convolution Layer,25 epochs.*

Berdasarkan hasil pengujian sistem,disimpulkan bahwa perubahan yang terjadi pada citra daun kentang mempengaruhi ketepatan dari proses klasifikasi,yaitu dengan merotasi citra, melakukan perubahan *brightness*,dan melakukan *shear range*.Pengujian 6 kali menggunakan rotasi 30° mendapatkan akurasi 50% dan untuk rotasi 60° mendapatkan akurasi sebesar 50%.Pengujian 6 kali penambahan *brightness* 40% mendapatkan akurasi 100% dan pengurangan *brightness* 40% mendapatkan akurasi 100%.Pengujian 6 kali melakukan *shear range* 15% mendapatkan akurasi 83,3% dan untuk perubahan *shear range* 30% mendapatkan akurasi 83,3%.

BAB V KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil pengujian penelitian yang telah dilakukan untuk mengklasifikasi penyakit pada daun kentang dengan metode *Convolutional Neural Network* (CNN) ini,dapat ditarik kesimpulan sebagai berikut :

1. Implementasi algoritma CNN dalam mengklasifikasikan citra daun kentang yang terserang penyakit dilakukan dengan mencari rancangan arsitektur model terbaik dengan melakukan kombinasi *hyperparameter* dan *Optimizer*. Berdasarkan hasil dari pengujian kombinasi *hyperparameter* pada setiap model CNN yang dirancang. Setiap *Hyperparameter* dan *Optimizer* yang diujikan memberi pengaruh yang besar pada model yang dihasilkan.Kombinasi paling optimal yang diperoleh dari hasil pengujian adalah dengan kombinasi *Optimizer* Adam, *Convolutional Layer* 5,dan *epochs* 100.
2. Hasil tingkat akurasi yang diperoleh dari data *testing* menggunakan model arstitektur terbaik dalam mengklasifikasikan citra jenis penyakit pada daun kentang yaitu sebesar 98,20 %.
3. Berdasarkan hasil pengujian sistem,disimpulkan bahwa perubahan yang terjadi pada citra daun kentang mempengaruhi ketepatan dari proses klasifikasi,yaitu dengan merotasi citra, melakukan perubahan *brightness*,dan melakukan *shear range*.Pengujian 6 kali menggunakan rotasi 30° mendapatkan akurasi 50% dan untuk rotasi 60° mendapatkan akurasi sebesar 50%.Pengujian 6 kali penambahan brightness 40% mendapatkan akurasi 100% dan pengurangan brightness 40% mendapatkan akurasi 100%.Pengujian 6 kali melakukan shear range 15% mendapatkan akurasi 83,3% dan untuk perubahan shear range 30% mendapatkan akurasi 83,3%.

5.2 Saran

Berdasarkan hasil yang diperoleh dari penelitian ini, saran yang diperlukan untuk perbaikan sistem maupun untuk penelitian selanjutnya adalah sebagai berikut :

1. Penelitian selanjutnya diharapkan dapat menggunakan dataset dengan jumlah yang lebih banyak, sehingga dapat menghasilkan performa model yang lebih baik, atau bisa dengan menggunakan data primer untuk pengujiannya.
2. Penelitian selanjutnya diharapkan dapat di kembangkan kedalam sebuah aplikasi yang digabungkan dengan *smartphone*.

DAFTAR PUSTAKA

Agarwal, M., Singh, A., Arjaria, S., Sinha, A., & Gupta, S. (2020). ToLeD: Tomato Leaf Disease Detection using Convolution Neural Network. *Procedia Computer Science*, *167*(1000), 293–301. https://doi.org/10.1016/j.procs.2020.03.225

Ahmad, U. (2005). Pengolahan Citra Digital Dan Teknik Pemrogramannya. Yogyakarta: Graha Ilmu.

Ali, M. M. et al. (2019b) 'Non-destructive techniques of detecting plant diseases: A review', Physiological and Molecular Plant Pathology. Elsevier Ltd, 108(June), p. 101426. doi: 10.1016/j.pmpp.2019.101426

Allaam, M. R. R., & Wibowo, A. T. (2021). Klasifikasi Genus Tanaman Anggrek Menggunakan Convolutional Neural Network. *E-Proceeding of Engineering*, *8*(2), 1153–1189.

Bauske, M. J., Robinson, A. P., & Gudmestad, N. C. (2018). Early Blight in Potato — Publications. *North Dekota State University Extention*, *June*, 8. https://www.ag.ndsu.edu/publications/crops/early-blight-in-potato

Brownlee, Jason. 2017. Gentle Introduction to the Adam Optimization Algorithm for Deep Learning. July 3. https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/.

C. Gershenson, "Artificial Neural Networks for Beginners," Networks, vol. cs.NE/0308, p. 8, 2003.

Chen, J., Zhang, D., Sun, Y., & Nanehkaran, Y. A. (2020). Using deep transfer learning for image-based plant disease identification. *Computers and Electronics in Agriculture*, *173*(March), 105393. https://doi.org/10.1016/j.compag.2020.105393

Chen, W., Sun, Q., Wang, J., Dong, J. J., & Xu, C. (2018). A Novel Model Based on AdaBoost and Deep CNN for Vehicle Classification. *IEEE Access*, *6*(c), 60445–60455. https://doi.org/10.1109/ACCESS.2018.2875525

Feng, J. and Lu, S. (2019) 'Performance Analysis of Various Activation Functions in Artificial Neural Networks', Journal of Physics: Conference Series, 1237, p. 022030. doi:10.1088/1742-6596/1237/2/022030

Fitriana, A., Hakim, L., & Marlina, M. (2020). The Effectiveness of Endophytic Fungi Origin of Potato Plant Roots in Coffee Skin Compost Media to Suppress Development of Potato Leaf Disease (Phytophtora infestans). *Budapest International Research in Exact Sciences (BirEx) Journal*, *2*(1), 101–105. https://doi.org/10.33258/birex.v2i1.760

G. A. Lujan-Moreno, P. R. Howard, O. G. Rojas, and D. C. Montgomery, "Design of experiments and response surface methodology to tune machine learning

hyperparameters, with a random forest case-study," Expert Syst. Appl., vol. 109, pp. 195–205, 2018, doi: 10.1016/j.eswa.2018.05.024.

Gonzalez, R. C. and Woods, R. E. (2002) Digital image processing. 2nd ed. Upper Saddle River, N.J: Prentice Hall.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press

Kim, P. (2017). MATLAB *deep learning : with machine learning, neural networks and artificial intelligence*. New York, NY: Apress.

Kozłowski, M., Górecki, P., & Szczypiński, P. M. (2019). Varietal classification of barley by convolutional neural networks. *Biosystems Engineering*, *184*, 155–165. https://doi.org/10.1016/j.biosystemseng.2019.06.012

Martinelli, F. et al. (2015) 'Advanced methods of plant disease detection. A review', Agronomy for Sustainable Development, 35(1), pp. 1–25. doi: 10.1007/s13593-014-0246-1.

Mayadewi, P., & Rosely, E. (2015). Prediksi Nilai Proyek Akhir Mahasiswa Menggunakan. *Seminar Nasional Sistem Informasi Indonesia*, *November*, 2–3.

Munir, Rinaldi, Pengolahan Citra Digital dengan Pendekatan Algoritmik, Bandung: Penerbit Informatika, 2004.

Nurfita, R. D., & Ariyanto, G. (2018). Implementasi Deep Learning Berbasis Tensorflow Untuk Pengenalan Sidik Jari. *Emitor: Jurnal Teknik Elektro*, *18*(01), 22–27. https://doi.org/10.23917/emitor.v18i01.6236

Oppenheim, D., & Shani, G. (2017). Potato Disease Classification Using Convolution Neural Networks. *Advances in Animal Biosciences*, *8*(2), 244–249. https://doi.org/10.1017/s2040470017001376

Patro, S. G. K., & sahu, K. K. (2015). Normalization: A Preprocessing Stage. *Iarjset*, 20–22. https://doi.org/10.17148/iarjset.2015.2305

Permadi, J., & Harjoko, A. (2015). Identifikasi Penyakit Cabai Berdasarkan Gejala Bercak Daun dan Penampakan Conidia Menggunakan Probabilistic Neural Network. *Semnaskit 20152*, 49–53.

Rahayu, T. K., & Mendes, J. A. (2021). *DETEKSI PENYAKIT TANAMAN RAMBUTAN BERDASARKAN CITRA DAUN MENGGUNAKAN FUZZY K - NEAREST NEIGHBOUR*. *10*(2), 41–46.

Rakhmawati, P. U., Pranoto, Y. M., & Setyati, E. (2018). *Klasifikasi Penyakit Daun Kentang Berdasarkan*. 1–8.

Rima Dias Ramadhani, Nur Aziz Thohari, A., Kartiko, C., Junaidi, A., Ginanjar Laksana, T., & Alim Setya Nugraha, N. (2021). Optimasi Akurasi Metode Convolutional Neural

Network untuk Identifikasi Jenis Sampah. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, *5*(2), 312–318. https://doi.org/10.29207/resti.v5i2.2754

Robinson, A. (2017). Late Blight in Potato. *ResearchGate*, *May*, 4. www.ag.ndsu.edu

Rozaqi, A. J., Sunyoto, A., & Arief, M. rudyanto. (2021). Deteksi Penyakit Pada Daun Kentang Menggunakan Pengolahan Citra dengan Metode Convolutional Neural Network. *Creative Information Technology Journal*, *8*(1), 22. https://doi.org/10.24076/citec.2021v8i1.263

Rozaqi, A. J., Sunyoto, A., & Arief, M. rudyanto. (2021). Deteksi Penyakit Pada Daun Kentang Menggunakan Pengolahan Citra dengan Metode Convolutional Neural Network. *Creative Information Technology Journal*, *8*(1), 22. https://doi.org/10.24076/citec.2021v8i1.263

Rumpf, T., Mahlein, A. K., Steiner, U., Oerke, E. C., Dehne, H. W., & Plümer, L. (2010). Early detection and classification of plant diseases with Support Vector Machines based on hyperspectral reflectance. *Computers and Electronics in Agriculture*, *74*(1), 91–99. https://doi.org/10.1016/j.compag.2010.06.009

S. Albawi, T. A. Mohammed and S. Al-Zawi, "Understanding of a convolutional neural network," 2017 International Conference on Engineering and Technology (ICET), 2017, pp. 1-6, doi: 10.1109/ICEngTechnol.2017.8308186.

Santosa, B., & Umam, A. (2018). *Data Mining dan Big Data Analytics*. Yogyakarta: Penebar Media Pustaka

Sari, I. P. (2016). Perancangan dan Simulasi Deteksi Penyakit Tanaman Jagung Berbasis Pengolahan Citra Digital Menggunakan Metode Color Moments dan GLCM. *Seminar Nasional Inovasi Dan Aplikasi Teknologi Di Industri (SENIATI)*, 215–220. https://doi.org/https://doi.org/10.36040/seniati.vi0.811

Schmidhuber, J. (2015). Deep Learning in neural networks: An overview. *Neural Networks*, *61*, 85–117. https://doi.org/10.1016/j.neunet.2014.09.003

Snadhika Jaya, T. (2018). Pengujian Aplikasi dengan Metode Blackbox Testing Boundary Value Analysis (Studi Kasus: Kantor Digital Politeknik Negeri Lampung). *Jurnal Informatika: Jurnal Pengembangan IT (JPIT)*, *03*(02), 45–48.

Steinbrener, J., Posch, K., & Leitner, R. (2019). Hyperspectral fruit and vegetable classification using convolutional neural networks. *Computers and Electronics in Agriculture*, *162*(April), 364–372. https://doi.org/10.1016/j.compag.2019.04.019

Suartika E. P, I Wayan, Wijaya Arya Yudhi, S. R. (2016). Klasifikasi Citra Menggunakan Convolutional Neural Network (Cnn) Pada Caltech 101. *Jurnal Teknik ITS*, *5*(1), 76. http://repository.its.ac.id/48842/

Taylor RJ, Pasche JS and Gudmestad NC 2008. Susceptibility of Eight Potato Cultivars to Tuber Infection by Phytophthora Erythroseptica and Pythium Ultimum and Its Relationship to Mefenoxam-Mediated Control of Pink Rot and

Tsany, A., & Dzaky, R. (2021). *Deteksi Penyakit Tanaman Cabai Menggunakan Metode Convolutional Neural Network*. *8*(2), 3039–3055. https://openlibrarypublications.telkomuniversity.ac.id/index.php/engineering/article/view/14701

Wahyu Nugraha, A. S. (2022). Hyperparameter Tuning pada Algoritma Klasifikasi dengan Grid Search. *Jurnal Sistem Informasi*, *11*(2), 391–401. https://doi.org/https://doi.org/10.32520/stmsi.v11i2.1750

Wasil, M., Harianto, H., & Fathurrahman, F. (2022). Pengaruh Epoch pada Akurasi menggunakan Convolutional Neural Network untuk Klasifikasi fashion dan Furniture. *Infotek : Jurnal Informatika Dan Teknologi*, *5*(1), 53–61. https://doi.org/10.29408/jit.v5i1.4393

Widiyanto, W. W. (2018). Analisa Metodologi Pengembangan Sistem Dengan Perbandingan Model Perangkat Lunak Sistem Informasi Kepegawaian Menggunakan Waterfall Development Model, Model Prototype, Dan Model Rapid Application Development (Rad). *Jurnal Informa Politeknik Indonusa Surakarta ISSN*, *4*(1), 34–40. https://doi.org/https://doi.org/10.46808/informa.v4i1.34

Wikarta, A., Pramono, A. S., & Ariatedja, J. B. (2020). Analisa Bermacam Optimizer Pada Convolutional Neural Network Untuk Deteksi Pemakaian Masker. *Seminar Nasional Informatika 2020 (SEMNASIF 2020)*, *2020*(Semnasif), 69–72.