

**KLASIFIKASI *IMAGE* TINGGI TANAMAN JAGUNG DENGAN
MENGUNAKAN ALGORITMA *CONVOLUTION NEURAL NETWORK*
(*CNN*)**

SKRIPSI

Diajukan Sebagai Salah Satu Syarat Untuk Mendapatkan
Gelar Sarjana Komputer (S.Kom) Program Studi Informatika



IKBAL

105841103620

**PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS MUHAMMADIYAH MAKASSAR
2024**



FAKULTAS TEKNIK

GEDUNG MENARA IQRA LT. 3

Jl. Sultan Alauddin No. 259 Telp. (0411) 866 972 Fax (0411) 865 588 Makassar 90221

Website: www.unismuh.ac.id, e_mail: unismuh@gmail.com

Website: <http://teknik.unismuh.makassar.ac.id>

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

PENGESAHAN

Skripsi atas nama Ikbal dengan nomor induk Mahasiswa 105841103620, dinyatakan diterima dan disahkan oleh Panitia Ujian Tugas Akhir/Skripsi sesuai dengan Surat Keputusan Dekan Fakultas Teknik Universitas Muhammadiyah Makassar Nomor : 0008/SK-Y/55202/091004/2024, sebagai salah satu syarat guna memperoleh gelar Sarjana Informatika pada Program Studi Informatika Fakultas Teknik Universitas Muhammadiyah Makassar pada hari Sabtu tanggal 26 Agustus 2024.

Panitia Ujian :

Makassar, 23 Safar 1446 H
28 Agustus 2024 M

1. Pengawas Umum

a. Rektor Universitas Muhammadiyah Makassar
Dr. Ir. H. Abd. Rakhim Nanda, ST., MT., IPU.

b. Dekan Fakultas Teknik Universitas Hasanuddin
Prof. Dr. Eng. Muhammad Isran Ramli, ST., MT.

2. Penguji

a. Ketua : Dr. Ir. Zahir Zainuddin, M.Sc.

b. Sekretaris : Rizki Yusliana Bakti, ST., MT.

3. Anggota

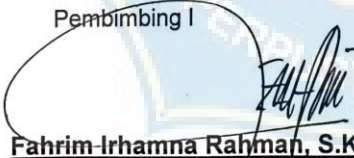
: 1. Lukman Anas, S.Kom., MT.

2. Muhyiddin A. M. Hayat, S.Kom., MT.

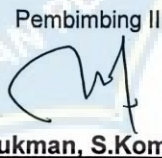
3. Titin Wahyuni, S.Pd., MT.

Mengetahui :

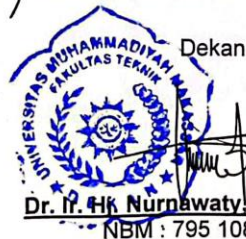
Pembimbing I


Fahrim-Irhamna Rahman, S.Kom., MT.

Pembimbing II


Lukman, S.Kom., MT.

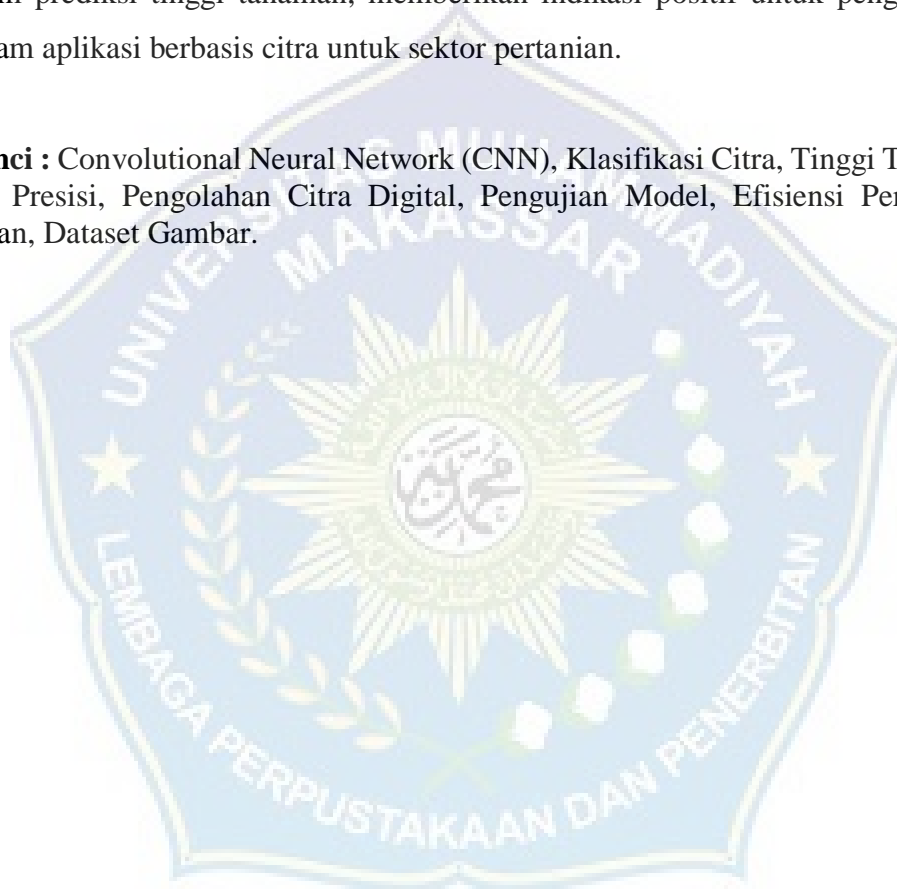
Dekan


Dr. Ir. Hik Nurnawaty, ST., MT., IPM.
NBM : 795 108

ABSTRAK

Penelitian ini mengimplementasikan algoritma Convolutional Neural Network (CNN) untuk klasifikasi citra tanaman jagung dengan fokus pada pengidentifikasian tinggi tanaman. Penelitian ini bertujuan untuk membantu petani dalam mendeteksi secara dini pertumbuhan tanaman jagung, yang dapat berkontribusi dalam pengelolaan pertanian yang lebih efisien. Model CNN yang dikembangkan berhasil mengklasifikasikan tinggi tanaman dengan akurasi maksimal mencapai 88% pada data uji dengan 1019 citra. Penggunaan CNN menunjukkan hasil yang cukup baik dalam prediksi tinggi tanaman, memberikan indikasi positif untuk pengembangan lebih lanjut dalam aplikasi berbasis citra untuk sektor pertanian.

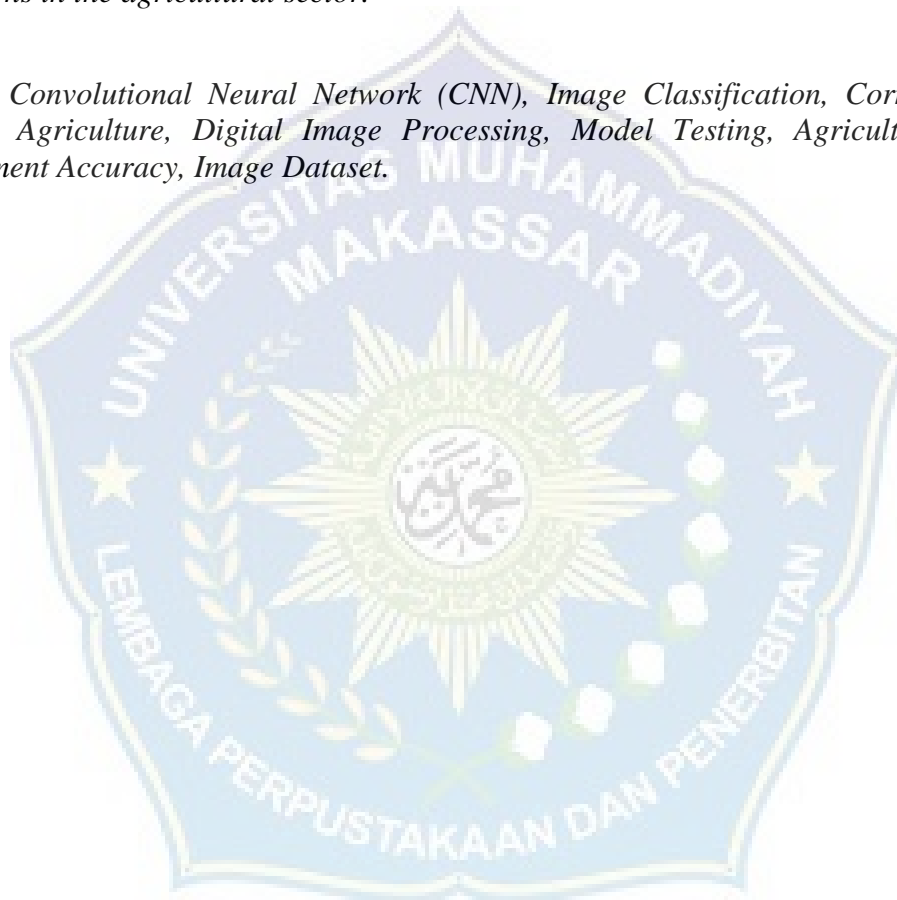
Kata Kunci : Convolutional Neural Network (CNN), Klasifikasi Citra, Tinggi Tanaman Jagung, Pertanian Presisi, Pengolahan Citra Digital, Pengujian Model, Efisiensi Pertanian, Akurasi Pengukuran, Dataset Gambar.



ABSTRACT

This research implements a Convolutional Neural Network (CNN) algorithm for classifying corn plant images, focusing on identifying plant height. The study aims to assist farmers in early detection of corn plant growth, contributing to more efficient agricultural management. The developed CNN model successfully classified plant height with a maximum accuracy of 88% on a test set of 1,019 images. The use of CNN demonstrated promising results in predicting plant height, indicating positive potential for further development of image-based applications in the agricultural sector.

Keyword : *Convolutional Neural Network (CNN), Image Classification, Corn Plant Height, Precision Agriculture, Digital Image Processing, Model Testing, Agricultural Efficiency, Measurement Accuracy, Image Dataset.*



KATA PENGANTAR

Assalamu'alaikum Warahmatullahi Wabarakatuh

Segala puji bagi Allah SWT yang selalu memberikan kita kenikmatan yang banyak yaitu nikmat iman, kesehatan, dan masih banyak nikmat lainnya. Sehingga tak ada satupun orang atau alat teknologi yang mampu menghitungnya. Sholawat beserta salam semoga tercurahkan kepada nabi Muhammad SAW sang revolusioner sejati yang menjadi suri tauladan seluruh umat, yang telah menyebarkan Islam berdakwah secara sembunyi-sembunyi dan secara terang-terang. Sehingga sampai detik ini kita bisa merasakan nikmatnya berislam.

Serta saya selaku penulis sangat bersyukur kepada Allah SWT, karena atas berkat dan rahmat-Nya saya dapat menyelesaikan Proposal Skripsi yang berjudul "Klasifikasi Image Tinggi Tanaman Jagung Menggunakan Algoritma Convolution Neural Network", serta dapat menyelesaikannya dengan tepat waktu.

Dalam penulisan Proposal kali ini saya selaku penulis ingin mengucapkan terima kasih kepada semua pihak telah membantu dan memberikan bimbingan baik secara langsung maupun tidak langsung. Dan pada kesempatan ini penulis mengucapkan terima kasih kepada :

1. **Allah SWT**, yang telah memberikan Beribu-ribu nikmat. Yaitu nikmat Islam, nikmat iman, nikmat kesehatan dan nikmat kesempatan untuk bisa menyelesaikan laporan ini.
2. **Teruntuk kedua orang tuaku** yang tercinta terutama untuk Ibu yang telah memberikan semangat perhatian, dan dukungan tanpa henti bagi saya, serta menguatkan saya dalam setiap doanya.
3. **Bapak Muhyiddin A M Hayat, S.Kom, M.T.**, selaku Ketua Program Studi Informatika Universitas Muhammadiyah Makassar yang telah memberikan saran.
4. Bapak Lukman Anas, S.Kom., M.T., sebagai dosen Pendamping akademik tempat minta saran dan masukan dari maba samapai sekarang.
5. **Bapak Fachrim Irhamna Rachman., S.kom., M.T.**, selaku Dosen Fakultas Teknik Program Studi Informatika dan juga Dosen Pembimbing I yang Telah banyak memberikan saran dan bimbingan.
6. **Bapak Lukman, S.Kom., M.T.**, selaku Dosen Pembimbing II yang telah banyak

memberikan bimbingan dan bantuannya.

7. Teman-Teman kelas B Program Studi Informatika Angkatan 2020 Terima kasih telah memberikan semangat kepada saya terutama atas nama Ikhlasul Amal, Aryansyah Iskandar yang menemani saya mengerjakan tugas akhirku.
8. Seluruh Staf yang telah membimbing dan memberi arahan kepada penulis sehingga dapat menyelesaikan penulisan proposal skripsi.
9. Terima Kasih juga kepada kak Irmawaty Hasyim, S.Sos. ,M.Sos & suami yang telah membantu saya dalam pengerjaan tugas akhirku.

Mengingat keterbatasan dan kemampuan penulis tentu proposal skripsi ini masih terdapat banyak kesalahan. Untuk itu penulis mengharapkan kritik dan masukan yang bermanfaat dari semua pihak. Semoga skripsi ini dapat bermanfaat bagi semua pihak.

Makassar, 06 Mei 2024

Peneliti



DAFTAR ISI

HALAMAN SAMPUK	i
ABSTRAK	ii
ABSTRACT	iii
KATA PENGANTAR	iv
DAFTAR ISI	v
DAFTAR TABEL	vii
DAFTAR GAMBAR	viii
Daftar Istilah	x
Bab I Pendahuluan	1
A. Latar belakang	1
B. Rumusan Masalah	2
C. Tujuan Penelitian	2
D. Manfaat Penelitian	3
E. Ruang Lingkup Penelitian	3
F. Sistematika Penyusunan	3
BAB II TINJAUAN PUSTAKA	5
A. Landasan Teori	5
a. Machine Learning	5
b. Artificial Intelligenci	5
c. Convolutional Neural Network (CNN)	6
d. Image Classification	6
e. Deep Learning	7
f. Jagung	7
g. TensorFlow	8
h. Confusion Matriks	8
B. Penelitian Terdahulu	11
C. Kerangka Pemikiran	17
BAB III METODE PENELITIAN	18
A. Waktu dan Tanggal Analisis	18
B. Alat dan Bahan	18
BAB IV HASIL DAN PEMBAHASAN	23
A. Pengukuran Tinggi Tanaman Secara Manual	23
B. Pengumpulan Dataset	23
C. Proses Pengolahan Dataset	27

D. Model Convolutional Neural Networkk (CNN).....	34
E. Pengujian Metode Convolutional Neural Network (CNN).....	38
F. Hasil Pengujian Dataset Gambar Tinggi Tanaman Jagung	42
BAB V KESIMPULAN DAN SARAN.....	47
A. Kesimpulan	47
B. Saram	47
DAFTAR PUSTAKA	48
LAMPIRAN.....	50





DAFTAR TABEL

Tabel 1.0 Penelitian Terdahulu.....	11
Table 1.1 Tabel Hasil Pengujian.....	42
Table 1.2 Tabel Hasil Pengujian.....	45



DAFTAR GAMBAR

Gambar 1. <i>Confusion matrix</i> (Samotana Zalukhu, 2023).....	11
Gambar. 2 kerangka pikir (sumber sendiri)	17
Gambar. 3 Alur Proses Learning	21
Gambar. 4 Pemeriksaan Alur Proses	22
Gambar. 5 Pengkuruan Tinggi Tanaman Secara Manual	23
Gambar. 5 Tinggi Tanaman Jagung 20cm.....	24
Gambar. 6 Tinggi Tanaman Jagung50cm.....	24
Gambar. 7 Tinggi Tanaman Jagung110cm.....	25
Gambar. 8 Tinggi Tanaman Jagung 20cm.....	25
Gambar. 9 Tinggi Tanaman Jagung 50cm.....	26
Gambar. 10 Tinggi Tanaman Jagung110cm.....	26
Gambar. 11 Kurva Akurasi.....	39
Gambar. 12 Kurva Loss	40
Gambar. 13 Tinggi Tanaman Jagung 20cm.....	41
Gambar. 14 Tinggi Tanaman Jagung 50cm.....	41



DAFTAR LAMPIRAN

Lampiran 1 Dataset Gambar Tanaman Jagung Ukuran 20cm.....	49
Lampiran 2 Dataset gambar tanaman jagung ukuran 50cm.....	50
Lampiran 3 Dataset gambar tanaman jagung ukuran 110cm.....	52
Lampiran 4 Source Code.....	53
Lampiran 5 Surat.....	60



DAFTAR ISTILAH

CNN	adalah singkatan dari "Convolutional Neural Network" atau dalam bahasa Indonesia dikenal sebagai "Jaringan Saraf Konvolusional". CNN adalah jenis arsitektur dalam deep learning yang khusus dirancang untuk memproses data yang memiliki struktur grid, seperti gambar dan data visual lainnya.
Pixel	Adalah singkatan dari "picture element" dalam bahasa Inggris. Ini adalah unit terkecil dalam representasi visual digital. Pixel adalah titik diskrit yang membentuk gambar atau tampilan pada layar komputer, perangkat mobile, kamera digital, atau perangkat elektronik lainnya yang mampu menampilkan gambar.
Machine Learning	Sebuah subbidang dari kecerdasan buatan yang berfokus pada pengembangan algoritma yang memungkinkan komputer untuk belajar dari data dan membuat prediksi atau keputusan tanpa diprogram secara eksplisit untuk setiap tugas.
Citra/Image	Citra adalah representasi visual dari objek atau adegan yang biasanya ditangkap oleh kamera atau dihasilkan melalui proses visual lainnya. Dalam konteks umum, citra merujuk pada gambar dua dimensi yang terdiri dari piksel-piksel, di mana setiap piksel mewakili nilai intensitas cahaya pada posisi tertentu dalam citra. Citra dapat merepresentasikan berbagai jenis informasi visual, seperti fotografi, lukisan, diagram, grafik, dan banyak lagi.
Deep Learning	Cabang dari pembelajaran mesin yang menggunakan jaringan saraf tiruan dengan banyak lapisan (deep neural networks) untuk menganalisis dan menemukan pola kompleks dalam data. Teknik ini sangat efektif dalam tugas-tugas seperti pengenalan gambar dan pemrosesan bahasa alami.
Flowchart	adalah representasi grafis dari urutan langkah-langkah atau proses dalam sebuah sistem, prosedur, atau algoritma. Flowchart digunakan untuk menggambarkan secara visual bagaimana suatu tugas atau proses dilakukan, sehingga memudahkan pemahaman dan komunikasi mengenai urutan langkah-langkah yang diperlukan.

TensorFlow	Sebuah platform open-source yang dikembangkan oleh Google untuk pembelajaran mesin. TensorFlow menyediakan berbagai alat dan perpustakaan untuk membangun dan melatih model pembelajaran mesin, termasuk model deep learning.
Supervised Learning	Jenis pembelajaran mesin di mana model dilatih menggunakan data yang telah diberi label. Model belajar dari pasangan input-output untuk memprediksi label yang benar untuk data baru.
Google Collab	adalah platform berbasis cloud yang disediakan oleh Google untuk menjalankan kode Python, terutama dalam konteks pembelajaran mesin dan pengembangan berbasis data. Ini adalah lingkungan yang sangat populer di kalangan peneliti, mahasiswa, dan praktisi di berbagai bidang yang memerlukan akses ke sumber daya komputasi yang kuat tanpa harus menginstal perangkat lunak atau membeli perangkat keras sendiri.
Phyton	adalah bahasa pemrograman tingkat tinggi yang sangat populer dan serbaguna. Diciptakan oleh Guido van Rossum dan pertama kali dirilis pada tahun 1991, Python dirancang dengan penekanan pada keterbacaan kode, sintaks yang jelas, dan efisiensi dalam pengembangan perangkat lunak. Ini membuatnya menjadi pilihan yang baik untuk berbagai aplikasi, mulai dari pengembangan web hingga ilmu data, pengembangan perangkat lunak, otomatisasi tugas, dan banyak lagi.
Data Augmentasi	Teknik yang digunakan untuk meningkatkan jumlah dan keragaman data pelatihan dengan melakukan transformasi pada data yang ada, seperti rotasi, pemotongan, flipping, dan perubahan warna. Ini membantu meningkatkan kinerja dan generalisasi model pembelajaran mesin.
Gradient Boosting	Sebuah teknik machine learning ensemble yang menggabungkan kekuatan beberapa model prediksi sederhana (seperti pohon keputusan) untuk membentuk model yang kuat. Teknik ini membangun model secara bertahap, setiap model baru mencoba untuk mengoreksi kesalahan dari model sebelumnya.
Reduksi Data	Proses seleksi dan penyederhanaan data mentah untuk memudahkan analisis. Melalui reduksi data, data yang tidak relevan

	atau berlebihan dapat dihilangkan, sehingga fokus pada informasi penting.
Triangulasi	Teknik untuk memverifikasi validitas dan keakuratan data dengan membandingkan beberapa sumber data atau sudut pandang yang berbeda. Ini membantu memastikan bahwa kesimpulan yang diambil berdasarkan data tersebut valid dan terpercaya.
Dataset	Kumpulan data yang digunakan untuk melatih, menguji, dan mengevaluasi model pembelajaran mesin. Dataset dapat berisi berbagai jenis data, seperti gambar, teks, atau angka, yang relevan dengan tugas yang akan diselesaikan oleh model.
Algoritma	Sekumpulan langkah atau prosedur yang digunakan untuk menyelesaikan suatu masalah atau melakukan tugas tertentu. Dalam konteks pembelajaran mesin, algoritma digunakan untuk membangun dan melatih model.
Komputasi	Proses penggunaan komputer untuk menyelesaikan berbagai tugas, termasuk perhitungan matematis, analisis data, dan pemrosesan informasi. Komputasi adalah dasar dari banyak teknologi dan aplikasi dalam pembelajaran mesin dan kecerdasan buatan.
Pengukuran Manual	Metode pengukuran tinggi tanaman yang dilakukan secara langsung oleh manusia dengan menggunakan alat pengukur seperti meteran atau penggaris

BAB I

PENDAHULUAN

A. Latar Belakang

Tanaman merupakan salah satu komponen penting dalam ekosistem dan memiliki peran signifikan dalam kehidupan manusia. Tinggi tanaman adalah salah satu parameter yang dapat memberikan informasi penting tentang pertumbuhan dan kesehatan tanaman. Pengukuran tinggi tanaman secara akurat memiliki nilai penting dalam berbagai aplikasi seperti pertanian, kehutanan, ekologi, dan ilmu lingkungan.

Pengukuran tinggi tanaman secara manual memerlukan waktu, tenaga, dan sumber daya yang signifikan. Selain itu, ketepatan pengukuran sering kali bergantung pada faktor subjektif yang dapat mengakibatkan kesalahan pengukuran. Oleh karena itu, pengembangan metode otomatis untuk menghitung tinggi tanaman menjadi sangat penting untuk meningkatkan efisiensi dan akurasi pengukuran. Dalam konteks ini, pemanfaatan Deep Learning, khususnya dalam bidang image processing atau pengolahan citra digital, menjadi relevan. Sistem image processing dirancang untuk membantu manusia dalam mengenali atau mengklasifikasi objek dengan cepat, tepat, dan dapat memproses banyak data sekaligus.

Pada bidang image processing terdapat beberapa algoritma yang dapat digunakan. Di antaranya adalah Naïve Bayes, Support Vector Machine, dan Neural Network. Salah satu algoritma yang sering digunakan adalah Neural Network. Neural Network dikembangkan berdasarkan cara kerja jaringan saraf pada otak manusia. Sejalan dengan perkembangan teknologi, maka, dikembangkan pula algoritma pengolahan citra digital. Salah satu pengembangan dari deep learning adalah Convolutional Neural Network. Pada tahun 1989, Yan LeCun, dkk mengembangkan model Neural Network dengan melakukan klasifikasi citra kode zip menggunakan kasus khusus dari Feed Forward Neural Network yang kemudian diberi nama Convolutional Neural Network (CNN). Metode Convolutional Neural Network memiliki hasil yang paling signifikan dalam pengenalan citra digital. Hal tersebut dikarenakan CNN diimplementasikan berdasarkan sistem pengenalan citra

pada visual cortex manusia.

Convolutional Neural Network (CNN) merupakan metode jaringan saraf mendalam, yang paling umum diterapkan dalam menganalisis gambar visual. Convolutional Neural Network juga disebut sebagai multilayer perceptron yang setiap neuronnya terhubung ke semua neuron dilapisan berikutnya. Namun disamping itu, Convolutional Neural Network juga mampu menemukan pola hierarkis dalam sebuah data dan mengumpulkan piksel yang lebih kompleks dari piksel yang lebih kecil dan juga lebih sederhana. Oleh karena itu, performa Convolutional Neural Network dalam keterhubungan dan kompleksitas piksel gambar sangat sesuai dan baik (Rasywir et al., 2020).

Berdasarkan latar belakang di atas bahwa penggunaan algoritma Convolution Neural Network dalam sistem klasifikasi image tanaman jagung dapat membantu mengetahui pertumbuhan tanaman jagung itu dengan mengidentifikasi gambar pada tanaman jagung dengan lebih efisien. Ini akan membantu petani jagung dalam menjaga tanaman mereka dan meningkatkan hasil panen mereka.

B. Rumusan Masalah

1. Bagaimana mengimplementasikan *Convolutional Neural Network (CNN)* untuk klasifikasi gambar tanaman jagung dengan akurasi maksimal?
2. Bagaimana teknologi klasifikasi tinggi berbasis citra, terutama menggunakan algoritma *CNN*, dapat diterapkan untuk mengidentifikasi secara akurat dalam mengklasifikasi tinggi tanaman menggunakan image?

C. Tujuan Penelitian

Mengimplementasikan *CNN* untuk Klasifikasi Gambar Tanaman Jagung

1. Mengimplementasikan model Convolutional Neural Network (CNN) untuk klasifikasi gambar tanaman jagung.
2. Mengembangkan dan menguji sistem klasifikasi tinggi tanaman jagung berbasis citra dengan menggunakan algoritma CNN, sehingga dapat memberikan hasil yang akurat.

D. Manfaat Penelitian

Dalam penelitian ini diharapkan akan memberikan manfaat sebagai berikut :

1. Membantu petani jagung dalam mendeteksi tinggi tanaman secara dini, yang dapat berkontribusi dalam pengelolaan pertanian yang lebih baik.
2. Meningkatkan produktivitas dan keberlanjutan sektor pertanian, khususnya bagi para petani baru.

E. Ruang Lingkup Penelitian

Mencakup batasan penelitian yang ditentukan yang dimaksudkan untuk menjaga penelitian pada jalur yang telah ditentukan. Pernyataan singkat dan tepat digunakan untuk menguraikan ruang lingkup penelitian.

1. Subjek Penelitian :

Penelitian ini difokuskan pada klasifikasi tinggi tanaman jagung menggunakan gambar atau citra digital.

2. Ketersediaan Data :

Membatasi penelitian, berdasarkan ketersediaan data yang relevan. Misalnya, penelitian dapat membatasi diri pada dataset tertentu yang mencakup variasi yang cukup untuk melatih dan menguji model CNN.

F. Sistematika Penyusunan

Sistem penulisan yang digunakan dalam penelitian ini adalah:

BAB I PENDAHULUAN

Bab ini menguraikan sejarah masalah, konseptualisasi, kendala, tujuan, kelebihan, dan metodologi penulisan.

BAB II TINJAUAN PUSTAKA

Bab ini memberikan penjelasan mengenai gagasan-gagasan yang melandasi penelitian serta referensi dan teori terkait yang menonjolkan gagasan kontroversial atau tidak logis (kerangka berpikir).

BAB III METODOLOGI PENELITIAN

Bab ini membahas metodologi penelitian yang meliputi perancangan sistem, prosedur pengujian, metodologi analisis information, lokasi dan waktu penelitian, serta instrumen dan bahan.

BAB IV HASIL DAN PEMBAHASAN

Bab ini memaparkan hasil penelitian serta pembahasan terhadap perancangan sistem, khususnya pengolahan data citra menggunakan metode Mobile Net dan implementasi model Machine Learning. Uji coba sistem juga dilakukan untuk memverifikasi hasil penelitian.

BAB V KESIMPULAN

Bab terakhir ini berisi rangkuman dan kesimpulan dari hasil penelitian, serta memberikan rekomendasi sesuai dengan keterbatasan yang ada dalam sistem yang dikembangkan.



BAB II

TINJAUAN PUSTAKA

A. Landasan Teori

a. *Machine Learning*

Penerapan teknologi komputer dan algoritma matematika yang menggunakan strategi pembelajaran dari information untuk menghasilkan prediksi masa depan dikenal sebagai pembelajaran mesin. Pelatihan dan pengujian adalah dua fase yang membentuk proses pembelajaran. Tujuan pembelajaran mesin adalah untuk membuat program komputer yang, dengan pengalaman, dapat bekerja lebih baik secara otomatis. Tiga kategori pembelajaran mesin telah diidentifikasi oleh penelitian terbaru: pembelajaran penguatan, pembelajaran tanpa pengawasan, dan pembelajaran yang diawasi.

Kecerdasan buatan mencakup pembelajaran mesin, yang banyak digunakan untuk mengatasi berbagai masalah. Tiga kategori berlaku untuk itu: pembelajaran penguatan, pembelajaran tanpa pengawasan, dan pembelajaran dengan pengawasan. Kategori-kategori ini dapat ditingkatkan lebih jauh lagi untuk mendapatkan tingkat akurasi dan presisi yang tinggi sekaligus menurunkan biaya komputasi dan meningkatkan kinerja. (Roihanet al., 2020).

b. *Artificial Intelligence*

Cabang ilmu komputer yang dikenal sebagai kecerdasan buatan (AI) berkaitan dengan pembuatan komputer cerdas yang mampu melakukan aktivitas yang biasanya membutuhkan kecerdasan manusia. Ilmu data, pembelajaran mesin, statistik, serta pendekatan dan teknik lain yang semuanya berada di bawah payung “belajar dari data” semuanya termasuk dalam kategori kecerdasan buatan (AI).

Perkembangan *Digital Twin Systems (DTS)* sangat dipengaruhi oleh kecerdasan buatan (AI). DTS memungkinkan pemantauan, analisis, dan pengambilan keputusan secara real-time dengan berfungsi sebagai versi virtual dari benda fisik. AI mendukung DTS dalam beberapa cara, termasuk pengambilan keputusan, pemodelan generatif, analisis information, pemodelan prediktif, serta pembuatan show dan pengoptimalan pembaruan. (Streib-

Emmert, 2023).

c. *Convolutional Neural Network* (CNN)

Menurut (Mudzakir & Arifin, 2022) salah satu metode yang populer dalam *Deep Learning* yaitu *Convolutional Neural Network (CNN)*. Menurut (Oktaviana et al., 2021) dan juga *Convolutional Neural Network* dinilai telah menunjukkan keunggulan yang luar biasa dalam berbagai penerapan di dunia nyata dibandingkan dengan sebagian besar pendekatan pada metode *Machine Learning* yang lain. *Convolutional Neural Network* juga termasuk salah satu jenis metode *supervised* yang cara kerjanya adalah menerima input berupa sebuah gambar dan juga bisa sangat cocok dalam pemrosesan input berupa citra. *Convolutional Neural Network* adalah suatu metode machine learning yang dikembangkan dari *Multi Layer Perceptron (MLP)* yang didesain untuk mengolah data 2D (Hawari et al., 2022).

Convolutional Neural Network (CNN) merupakan metode jaringan saraf mendalam, yang paling umum diterapkan dalam menganalisis gambar visual. *Convolutional Neural Network* juga disebut sebagai *multilayer perceptron* yang setiap neuronnya terhubung ke semua neuron dilapisan berikutnya. Namun disamping itu, *Convolutional Neural Network* juga mampu menemukan pola hierarkis dalam sebuah data dan mengumpulkan piksel yang lebih kompleks dari piksel yang lebih kecil dan juga lebih sederhana. Oleh karena itu, performa *Convolutional Neural Network* dalam keterhubungan dan kompleksitas piksel gambar sangat sesuai dan baik (Rasywir et al., 2020).

Menurut (Gogul, I. & Kumar, V. S. 2017) dalam model arsitektur pada metode *Convolutional Neural Network* juga dapat dikembangkan dengan *transfer learning Convolutional Neural Network* model menggunakan *pre-Trained* atau sebelumnya sudah dilakukan proses pelatihan oleh orang lain atau peneliti dengan menggunakan dataset yang sangat besar salah satunya menggunakan dataset *imagenet* (Oktafanda, 2022).

d. *Image Classification*

Mengklasifikasikan piksel atau objek dalam suatu gambar ke dalam beberapa kelas atau kategori merupakan proses sulit yang dikenal dengan

klasifikasi gambar. Ini adalah tugas penting untuk penelitian penginderaan jauh karena banyak aplikasi lingkungan dan sosial ekonomi bergantung pada temuan kategorisasi tersebut.

Ada beberapa metode dan strategi untuk mengklasifikasikan gambar. Sampel pelatihan digunakan dalam teknik klasifikasi terawasi untuk menetapkan kelas tutupan lahan dan menyediakan data referensi bagi pengklasifikasi yang akan dilatih. Jaringan saraf tiruan, pengklasifikasi pohon keputusan, jarak minimal, dan kemungkinan maksimum adalah beberapa contoh pengklasifikasi yang diawasi. Lu dan Weng (2007).

e. *Deep learning*

Tujuan pembelajaran mendalam, salah satu cabang pembelajaran mesin, adalah untuk melatih jaringan saraf tiruan (JST) berlapis untuk mengenali dan mengekstrak pola dan representasi rumit dari informasi. Ini didasarkan pada arsitektur ANN asli, yang mencakup teknik optimasi dan aktivasi bersama dengan struktur multilayer. Pembelajaran mendalam telah banyak digunakan di berbagai bidang, seperti malware, botnet, dan deteksi intrusi. Sistem berbasis pembelajaran mendalam tidak terlalu bergantung pada aturan yang telah ditentukan atau tanda tangan serangan dalam hal deteksi intrusi. Sebaliknya, dengan memeriksa karakteristik fitur lalu lintas jaringan (NTF) yang dapat diukur, mereka menggunakan pelatihan informasi empiris untuk mengenali serangan jaringan. (Toldinas dan lainnya, 2021).

f. *Jagung*

Tanaman jagung (*Zea mays*) adalah salah satu jenis tanaman pangan yang sangat penting secara global. Jagung merupakan salah satu jenis tumbuhan langka yang berasal dari tumbuhan famili Poaceae.

Jagung adalah salah satu tanaman pangan penghasil karbohidrat yang terpenting di dunia, selain gandum dan padi. Tanaman jagung merupakan salah satu jenis tanaman pangan biji-bijian dari keluarga rumput-rumputan. Berasal dari Amerika yang tersebar ke Asia dan Afrika melalui kegiatan bisnis orang-orang Eropa ke Amerika. Sekitar abad ke-16 orang Portugal menyebarkan ke Asia termasuk Indonesia. Orang Belanda

menamakannya mais dan orang Inggris menamakannya corn. (Maramba & Sidiq Purnomo, 2024).

Tanaman jagung dalam arti khusus merupakan salah satu jenis pangan yang sangat bermanfaat bagi kehidupan manusia bahkan hewan. Jagung adalah makanan kedua yang disantap setelah makan di Indonesia. Sebaliknya, berdasarkan pola pangan global, jagung dikonsumsi empat kali setelah makan dan snack. Masyarakat umum memanfaatkan penyamakan kulit dalam beberapa cara, seperti: tepung jagung (maizena), minyak jagung, bahan pangan, serta sebagai pakan ternak dan kegunaan lainnya. (Dora Fatma Nurshanti et al., 2019).

g. *TensorFlow*

Google membuat framework pembelajaran mesin sumber terbuka TensorFlow. TensorFlow menawarkan ekosistem alat, perpustakaan, dan sumber daya yang luas untuk membuat dan mengimplementasikan demonstrasi pembelajaran mesin. Fleksibilitas, skalabilitas, dan dukungan komunitas TensorFlow yang luas menjadikannya alat yang populer di berbagai bidang, termasuk sains, bisnis, dan pemerintahan.

Kapasitas *TensorFlow* untuk mengelola pelatihan dan inferensi yang didistribusikan secara besar-besaran adalah salah satu karakteristik utamanya. Menggunakan API tingkat rendah untuk kontrol yang lebih presisi atau API tingkat tinggi seperti Keras, pengguna dapat membuat dan melatih model pembelajaran mendalam yang canggih. Peneliti dan pengembang dapat bereksperimen dengan model dan metodologi alternatif berkat dukungan komprehensif TensorFlow terhadap operasi dan arsitektur jaringan. (David dan lainnya, 2020).

h. *Confusion Matrics*

Matriks konfusi adalah tabel yang menunjukkan keluaran model klasifikasi dengan data respons yang terpantau dan akurat. Berdasarkan skenario informasi yang diantisipasi atau diklasifikasikan, akurasi, presisi, F-Score, dan beberapa variabel lainnya dapat dihitung untuk demonstrasi klasifikasi saat ini dalam tabel yang dihasilkan. Tabel ini memiliki empat isi, yaitu sebagai berikut

(Srihari & Kumar, 2022).

1. Skenario Genuine Positive (TP) terjadi ketika show memprediksi hasil yang positif (Genuine) dan hasil sebenarnya juga positif (Genuine).
2. Ketika demonstrate mengklasifikasikan information sebagai negatif (Untrue) dan hasil sebenarnya juga negatif (Wrong), keadaan tersebut disebut Genuine Negative (TN).
3. Situasi yang disebut Untrue Positive (FP) terjadi ketika demonstrate menginterpretasikan information sebagai positif (Genuine) padahal, kenyataannya, responsnya adalah Untrue (Untrue).
4. Skenario yang disebut Untrue Negative (FN) terjadi ketika show menafsirkan information sebagai negatif (Untrue) meskipun hasil sebenarnya adalah positif (Genuine).

Akurasi, presisi, review, dan f-score merupakan kriteria yang akan digunakan sesuai dengan informasi dari matriks konfusi. Akurasi sistem diukur dari seberapa baik sistem dapat mengelompokkan totalnya. Presisi didefinisikan sebagai rasio jumlah add up to information yang dikategorikan positif terhadap jumlah information dalam kategori positif yang diklasifikasikan dengan benar oleh sistem. Pengukuran sistem untuk information yang dikategorikan positif secara akurat disebut review. F-Score adalah metrik yang digunakan untuk menghitung perolehan dan presisi secara bersamaan. Rata-rata harmonik akurasi dan recall digunakan oleh F-Score.

Cara menentukan akurasi, khususnya

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \dots\dots\dots (1)$$

Ketika semua information memiliki nilai positif dan dikategorikan dengan benar oleh sistem, information yang lengkap akan dibandingkan untuk mendapatkan presisi. Bagaimana menghitung presisi, khususnya

$$Precision = \frac{TP}{TP + FP} \dots\dots\dots (2)$$

Penarikan kembali adalah pengukuran data yang kategorisasinya bernilai positif dan dianggap akurat oleh sistem. Bagaimana penarikan kembali dihitung, secara spesifik.

$$Recall = \frac{TP}{TP + FN} \dots\dots\dots (3)$$

F-Score menghitung ukuran perolehan dan presisi secara bersamaan. Cara menentukan F-Score khususnya.

$$F-Score = 2 \cdot \frac{precision \cdot recall}{precision + recall} \dots\dots\dots (4)$$



		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

Gambar 1. *Confusion matrix* (Samotana Zalukhu, 2023).

B. Penelitian Terdahulu

Tabel 1.0 Penelitian Terdahulu

Peneliti	Tujuan / Kasus	Metode / Algoritma	Hasil
Mahdiyah U JAMI: Jurnal Ahli Muda Indonesia (2023) 4(1) 61-69	Pengklasifikasian kualitas cabai menggunakan <i>Gradient Boosting</i>	<i>ALGORITMA GRADIEN BOOSTING</i>	Gradien Boosting cukup bagus untuk klasifikasi kualitas cabai dengan banyak kelas ada 5 dengan rata-rata nilai precision, recall, F-Score dan akurasinya adalah sekitar 69,7%, 69,1% 69,7%, dan $\pm 76\%$.

<p>Teknik Elektro J, Wega Intyanto G (2021)</p>	<p>Penulis mencoba melakukan penelitian klasifikasi citra bunga dengan membuat model arsitektur deep learning, yaitu arsitektur pada CNN yang memiliki kelebihan dalam hal ekstraksi fitur secara otomatis.</p>	<p><i>Deep Learning: CNN (Convolution Neural Network)</i></p>	<p>Hasilmenyatakanketika dilakukan iterasi sebanyak 50 kali bahwa pada Arsitektur Inilai akurasi-nyayaitu 0.62 sedangkan arsitekturIIdengan menggunakanVGG16 memiliki akurasi 0.8 dandaridinyatakan lebih baik dibandingkan dengan arsitektur I.</p>
<p>Wolas N, Endah M (2022)</p>	<p>Penelitian ini dilakukan untuk membuat sebuah aplikasi bernama “Aplikasi Deep Learning Menggunakan Algoritma Convolutional Neural Network (CNN) untuk Klasifikasi Burung dari Keluarga Acciptridae” untuk mengklasifikasikan beberapa spesies dalam keluarga Acciptridae yang terancam punah yaitu Elang Peregrine Jambul , Burung Sikep-Madu Asia, Elang Kuskus, dan Elang Jawa. Tujuan dari aplikasi ini</p>	<p><i>Algoritma Convolutional Neural Network(CNN)</i></p>	<p>aplikasi penggunaan metode Convolutional Neural Network (CNN) untuk mengkategorikan burung dalam keluarga Acciptridae. Empat jenis burung terancam punah yang masuk dalam klasifikasi adalah elang madu asia, elang jawa, elang peregrine jambul, dan elang kuskus. Foto tahun 1968 secara keseluruhan, dari empat jenis burung yang dimanfaatkan,</p>

	<p>adalah untuk membantu petugas konservasi mengidentifikasi jenis burung dalam keluarga Accipitridae dan membantu masyarakat meminimalisir perdagangan ilegal empat jenis burung dalam keluarga tersebut.</p>		<p>terbagi menjadi 80% data latih (1576 gambar) dan 20% data validasi (329 gambar). CNN mampu mengkategorikan foto empat spesies berbeda dengan akurasi 100% untuk data pelatihan dan akurasi 87% untuk data validasi.</p>
<p>Sri Rahmadhani U, Lysbetti Marpaung N (2023)</p>	<p>Suatu penelitian yang berdasarkan Sort untuk mengklasifikasi jamur yang dapat diimakan dan beracun. Jamur yang digunakan dalam penelitian hanya ada tiga jenis, yaitu Boletus (konsumsi), Boletus (beracun), Ganoderma (konsumsi), Ganoderma (beracun), dan Russula (konsumsi) yang akan dibagi dalam beberapa kategori. Penggunaan keempat kelas tersebut disebabkan oleh</p>	<p><i>Algoritma Convolutional Neural Network(CNN)</i></p>	<p>Klasifikasi Jamur berdasarkan Kelas yang dilakukan menggunakan metode CNN untuk informasi dengan grafis 1200 warna. Tiga kelas digunakan dalam penelitian ini.</p> <p>Ini kemudian dibagi menjadi delapan kategori: makanan (Boletus), makanan (Beracun), Ganoderma (Konsumsi), makanan (Ganoderma),</p>

sedikitnya jenis buah yang dikonsumsi dan mempunyai kesamaan amorf dalam arti amorf. Bagaimanapun, metode yang digunakan dalam penelitian ini disebut Convolutional Neural Networking (CNN).

makanan (Russula), dan makanan (Beracun). Setelah pelatihan menggunakan pertunjukan yang menggunakan parameter ukuran tandan 64 dan umur, akurasi persiapan mencapai 89%, dengan tingkat validitas 82%. Di masa lalu Pengujian demonstrasi memperoleh skor rata-rata 76%. Setelah diimplementasikan, tampilan web tersebut dapat mengklasifikasikan gambar berwarna berdasarkan algoritma pengurutan yang andal. Terdapat beberapa kesalahan dalam penelitian ini, dan masih perlu dilakukan perbaikan. Untuk meningkatkan keakuratan demonstrasi, informasi

dapat ditambahkan atau data yang lebih baik dapat menggantikan gambar lama. Selain itu, teknik pembelajaran mendalam seperti Exchange Learning diterapkan untuk meningkatkan akurasi arsitektur yang sudah tersedia.

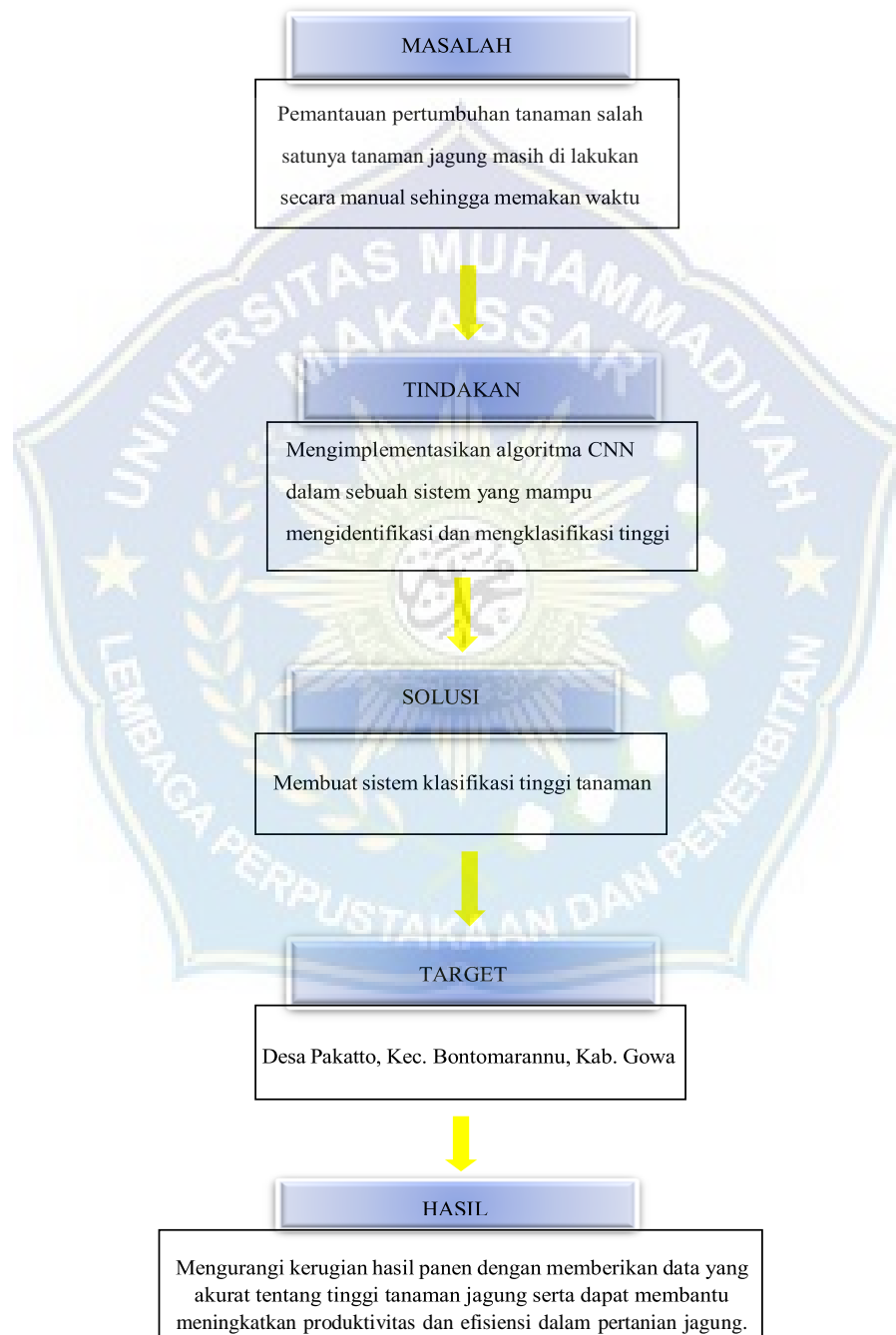
<p>Sely Wita D, Yanti Liliana D</p>	<p>Penggunaan CNN untuk mengklasifikasikan identitas menggunakan</p>	<p><i>Algoritma Convolutional</i></p>	<p>Berdasarkan hasil analisis dan interpretasi klasifikasi</p>
<p>JURTI (2022)</p>	<p>telapak tangan diharapkan dapat memberikan hasil yang akurat dengan tingkat kesalahan yang rendah.</p>	<p><i>Neural Network(CNN)</i></p>	<p>identitas menggunakan citra telapak tangan menggunakan Convolutional Neural Arrange (CNN) dan informasi identitas dari 26 nama, diperoleh perkiraan akurasi klasifikasi sebesar 98%, akurasi review sebesar 99 %, dan skor F1 adalah 98%. Berdasarkan nilai akurasi tersebut di atas, dapat disimpulkan</p>

bahwa klasifikasi identitas dengan metode CNN mempunyai nilai akurasi yang baik.



C. Kerangka Pemikiran

Kerangka pemikiran dalam proposal ini adalah memadukan teori klasifikasi image dan deep learning dengan studi pertanian jagung milik warga desa pakatto kabupaten gowa. Dengan demikian, akan didapatkan cara melihat pertumbuhan atau ukuran tinggi tanaman jagung yang mempermudah para petani nantinya.



Gambar. 2 kerangka pikir (sumber sendiri)

BAB III

METODE PENELITIAN

A. Waktu dan Tanggal Analisis

Dua faktor yang sangat penting yang harus dipertimbangkan selama proses penelitian adalah lokasi dan Durasi penelitian. Kedua faktor ini berperan penting dalam memastikan validitas dan keakuratan temuan penelitian.

Lokasi penelitian merupakan suatu lokasi atau objek tertentu yang akan dijadikan tempat dilakukannya suatu penelitian. Pemilihan lokasi penelitian merupakan hal yang penting dalam proses penelitian, karena akan memperlancar tugas peneliti. Tempat pengumpulan data untuk penelitian ini, di lakukan di Desa Pakatto, Kec. Bontomarannu, Kab. Gowa.

Adapun waktu penelitian yang akan di selesaikan dimulai pada bulan Mei 2024 sampai proses pengumpulan data selesai.

B. Alat dan Bahan

Diantara alat dan bahan yang digunakan dalam penelitian ini adalah sebagai berikut :

Persyaratan Perangkat Keras (*Hardware*)

- Laptop Acer Aspire 5 a
- RAM 8,00 GB
- OS Windows 11
- Intel Core i3
- HP Vivo Y21
- RAM 4,00GB
- Tripod
- Meteran

Persyaratan Perangkat Lunak (*Software*)

1. Google Kolaborasi
2. Kumpulan data
3. Operasi Sistem Windows 11

4. Honeyview

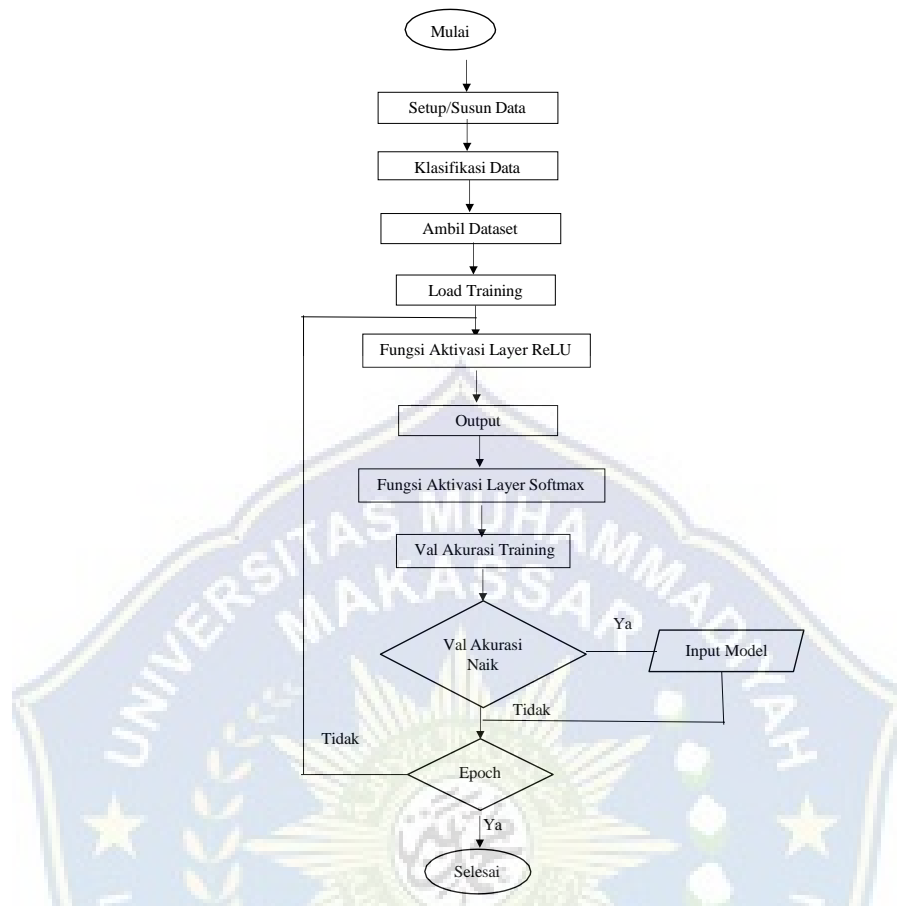
a. *Preprocessing*

Mempersiapkan data mentah untuk pemrosesan sistem adalah tujuan dari proses prapemrosesan. Dua dari tiga kelas foto yang menunjukkan ukuran tanaman jagung dipilih untuk langkah pra-pemrosesan penelitian ini. Setelah terpilih, kumpulan gambar buah-buahan tersebut Akses Google Drive yang berfungsi sebagai media pengembangan Google Collaboratory.

b. *Learning*

Tujuan dari proses pembelajaran yang disebut juga pelatihan ini adalah dengan menggunakan model CNN yang telah dikembangkan dengan kemampuan mendeskripsikan dan menganalisis gambar jaguar yang telah diberi indeks berdasarkan kelasnya. Dua kelas ukuran tanaman jagung terdiri dari 400 foto yang merupakan kuantitas data gambar yang digunakan untuk proses pembelajaran. Gambar kelas uji sebanyak 238 foto dari dua kelas tanaman jagung digunakan sebagai data validasi. Sepuluh epoch digunakan untuk menyelesaikan proses pembelajaran. Model CNN yang menghasilkan nilai indeks akurasi terbesar untuk mengkategorikan foto tinggi tanaman akan dipilih setelah 10 epoch.

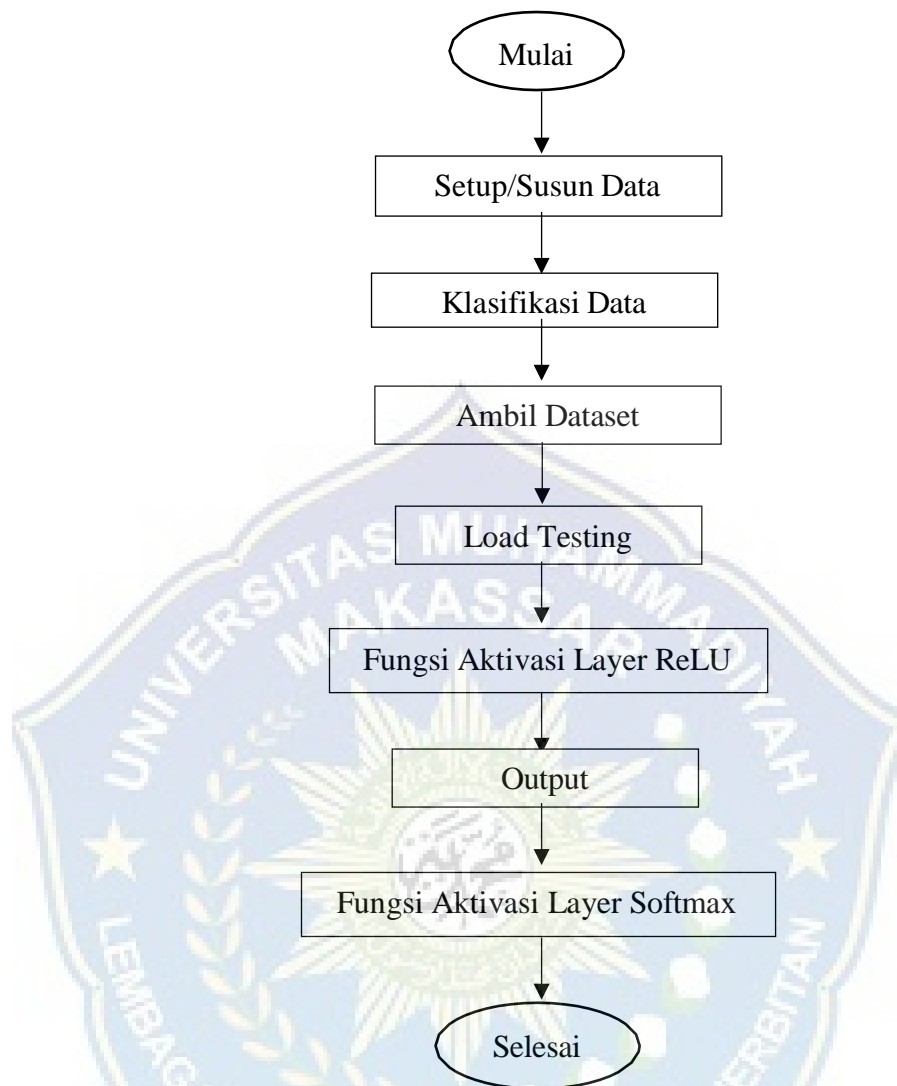
Dalam metode pembelajaran ini, variabel akurasi $> 0,9924$ dan kerugian $< 0,0221$ digunakan sebagai tujuan. Pada Gambar 3 ditampilkan proses pembelajaran.



Gambar. 3 Alur Proses Learning

c. Testing

Langkah terakhir dalam keseluruhan sistem penelitian adalah prosedur pengujian. Indeks model CNN yang dilatih dievaluasi sebagai bagian dari prosedur pengujian untuk menentukan seberapa akurat klasifikasinya. Gambar 4 menggambarkan alur prosedur pengujian.



Gambar. 4 Pemeriksaan Alur Proses

Proses pengujian diawali dengan resize (mengubah ukuran) gambar menjadi 256 x256 piksel. CNN melakukan klasifikasinya pada gambar masukan setelah diubah ukurannya. Gambaran uji kemungkinan dan kategorisasi kelas tanaman jagung merupakan hasil prosedur klasifikasi CNN. Dalam penelitian ini digunakan metode pengujian sebanyak 200 foto uji dengan rincian 100 gambar dari masing-masing dua kelas gambar tanaman jagung.

d. Teknik Pengujian Sistem

Peneliti menggunakan Confusion Matrix sebagai teknik pengujian untuk menguji akurasi dari model yang dibuat, teknik ini membagi antara total jumlah

dari data dengan total jumlah data yang bisa terdeteksi. Perhitungan akurasinya dapat diitung menggunakan rumus (1).

$$\text{Akurasi} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

e. Penarikan Kesimpulan

Membuat kesimpulan adalah langkah terakhir dalam proses analitis. Pada titik ini, peneliti mulai menyelidiki hipotesis penelitian, korelasi sebab-akibat, dan arti dari variabel yang digunakan. Untuk mencari solusi atas permasalahan yang dihadapi, data yang dikumpulkan dibandingkan satu sama lain.



BAB IV

HASIL DAN PEMBAHASAN

Pada bab ini, akan disajikan hasil dan pembahasan mengenai tinggi tanaman jagung menggunakan deteksi citra di Desa Pakatto, Kabupaten Gowa. Untuk hasil eksperimen berupa dataset yang meliputi data testing, dan data training, dengan masing-masing di isi dengan dua penyakit yaitu penyakit kuning dan penyakit layu bakteri. Dimana jumlah dataset gambar yaitu 1019 gambar.

A. Pengukuran Tinggi Tanaman Secara Manual

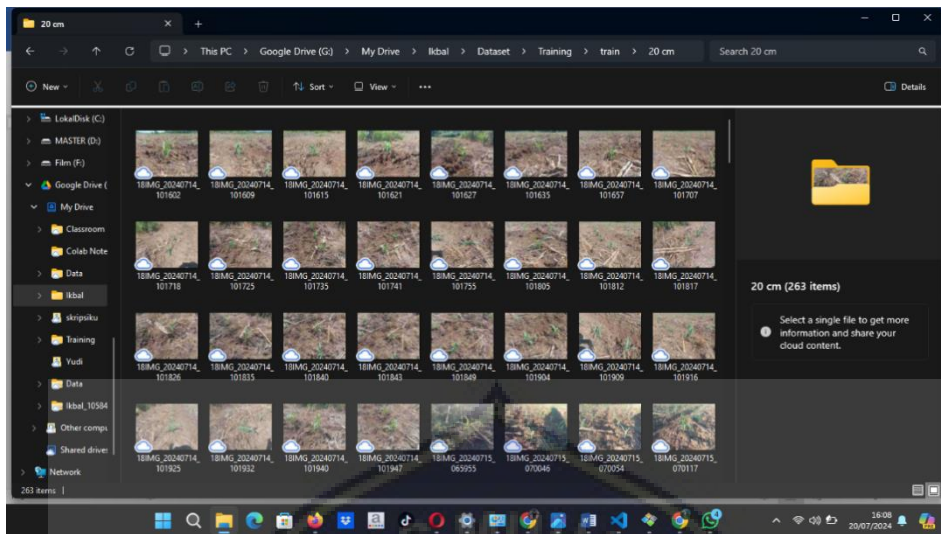
mengukur tinggi tanaman jagung secara manual di salah satu lahan milik petani yang berada di desa pakatto, kabupaten gowa. Pengukuran secara manual ini memiliki akurasi 100% dengan menggunakan alat pengukur tukang bangunan dengan mengukur dengan tinggi tanaman 20 cm, 50 cm dan 110 cm.



Gambar 5. Pengukuran Tinggi Tanaman Secara Manual

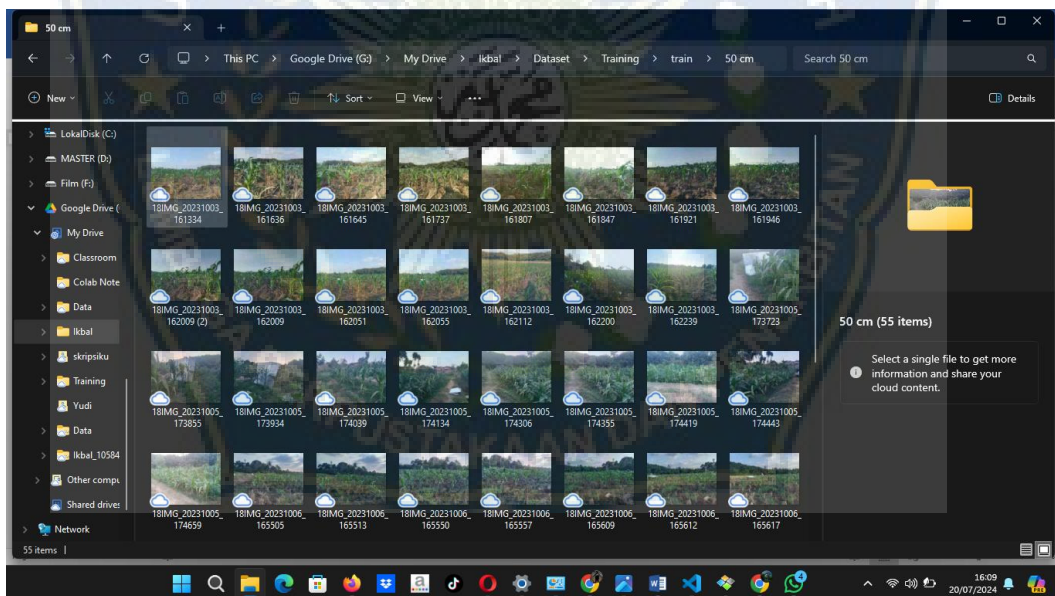
B. Pengumpulan Dataset

Dataset yang digunakan dalam penelitian ini merupakan data gambar batang untuk tinggi tanaman yang diperoleh dari lahan pertanian warga Desa Pakatto, Kabupaten Gowa sebanyak 1019 data, dataset ini kemudian di bagi menjadi data *testing*, dan data *training*. Citra dari tinggi tanaman berdasarkan batang yang diambil yaitu adalah citra berwarna dan memiliki *size* 100 dengan *pixel* 256. Dalam penelitian ini menggunakan 3 kelas, yaitu tinggi tanaman 20cm, 50cm dan 110cm. Adapun hasil pengumpulan data di Desa Pakatto, Kabupaten Gowa antara lain sebagai berikut:



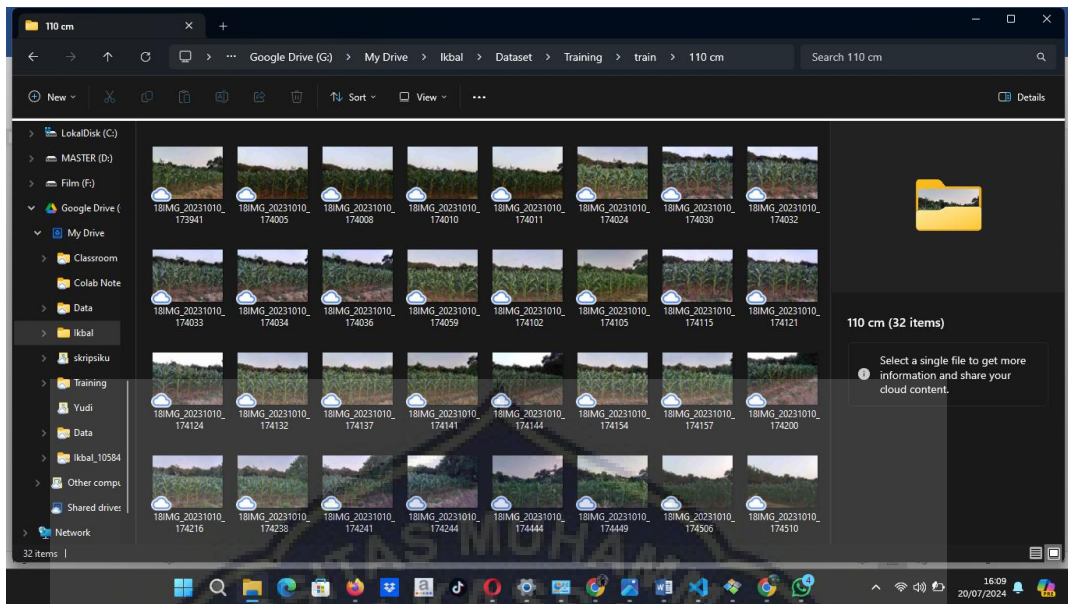
Gambar 6. Tinggi Tanaman 20 CM

Gambar diatas adalah gambar dataset *training* pada kelas tinggi tanaman 20cm.



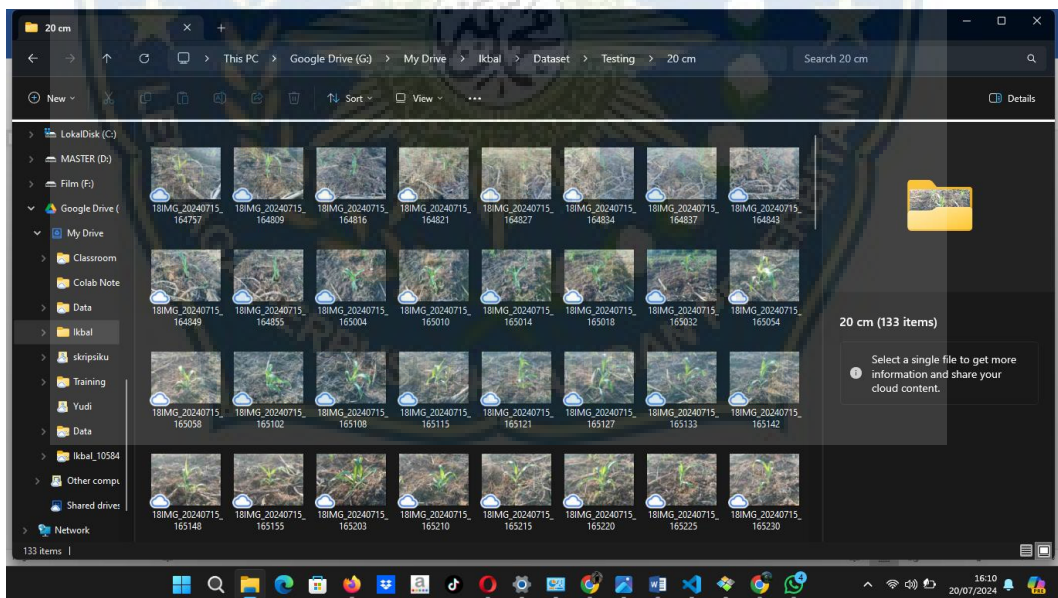
Gambar 7. Tinggi Tanaman 50 CM

Gambar diatas adalah gambar dataset *training* pada kelas tinggi tanaman 50cm.



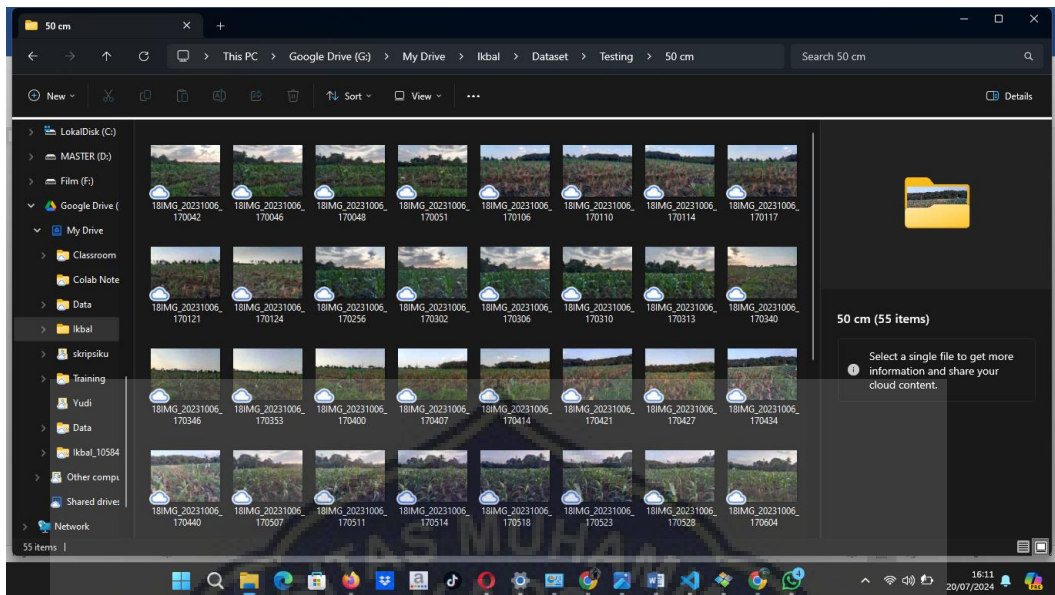
Gambar 8. Tinggi Tanaman 110 CM

Gambar diatas adalah gambar dataset *training* pada kelas tinggi tanaman 110cm.



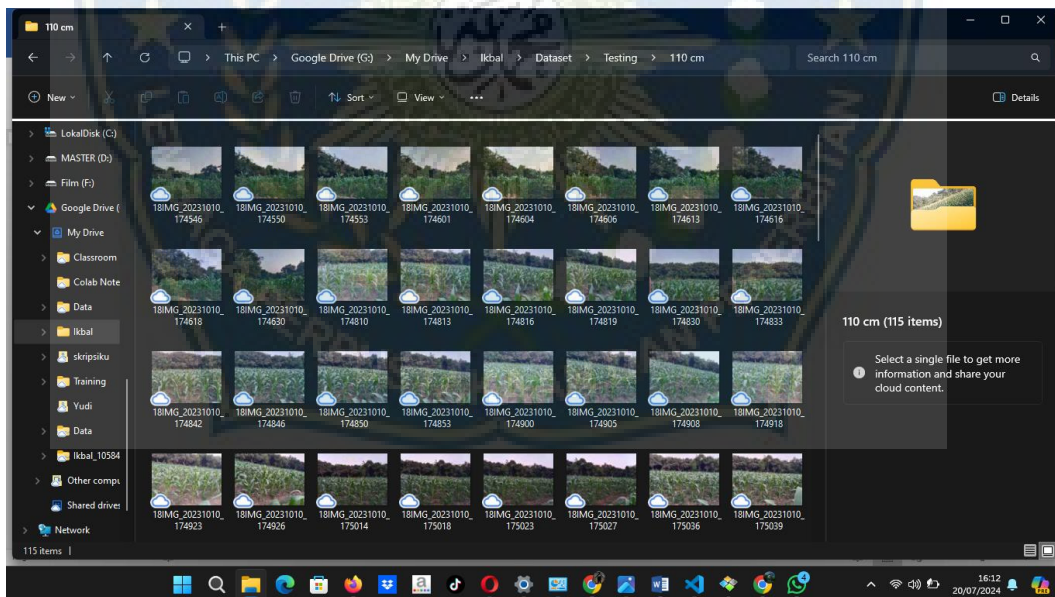
Gambar 9. Tinggi Tanaman 20 CM

Gambar diatas adalah gambar dataset *testing* pada tinggi tanaman 20cm.



Gambar 10. Tinggi Tanaman 50 CM

Gambar diatas adalah gambar dataset *testing* pada kelas tinggi tanaman 50cm.



Gambar 11. Tinggi Tanaman 110 CM

Gambar diatas adalah gambar dataset *testing* pada tinggi tanaman 110cm.

A. Memuat Dataset

Pada tahap ini digunakan dalam lingkungan *Google Colab* untuk

menghubungkan *Google Drive* ke dalam lingkungan tersebut. Ini memungkinkan untuk mengakses file dan folder yang ada di *Google Drive* langsung dari lingkungan *Colab* tanpa perlu mengunggah atau mengunduh secara manual.

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

C. Proses Pengelolaan Dataset

1. Kemudian mengimpor berbagai pustaka dan modul yang digunakan dalam pengembangan model jaringan saraf tiruan (neural network) menggunakan TensorFlow dan Sklearn.

```
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelBinarizer

from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import BatchNormalization
from tensorflow.keras.layers import Dense, Dropout,
Flatten, Conv2D, MaxPooling2D, Activation

from tensorflow.keras import backend as K
from tensorflow.keras.preprocessing import image
from tensorflow.keras.preprocessing.image import
ImageDataGenerator
from tensorflow.keras.optimizers import Adam, SGD
from tensorflow.keras.preprocessing.image import
img_to_array

from tensorflow.keras.callbacks import EarlyStopping
from tensorflow.keras.callbacks import ModelCheckpoint

import numpy as np
import matplotlib.pyplot as plt
import cv2
import pickle
from os import listdir
```

2. Pada tahap ini terkait dengan pengaturan parameter dan konfigurasi yang di perlukan untuk melatih sebuah model jaringan saraf tiruan (*neural network*) dengan dataset gambar.

Dalam hal ini epoch dalam pengelolah citra adalah literasi dalam model atau system. Dimana setiap epoch melibatkan proses memasukkan data, pelatihan dalam model, menghitung hasil prediksi, dan membandingkan hasil tersebut dengan data yang ada. Tujuan menggunakan banyak epoch adalah untuk memperbaiki kinerja model atau sistem seiring dengan berjalannya waktu. Dimana pada source code dibawah menggunakan epoch 10 agar prediksi yang diinginkan tercapai.

Dan batch size mengacu pada jumlah data pelatihan yang diberikan kepada model dalam satu iterasi. Ukuran batch size dapat mempengaruhi waktu pelatihan dan menggunakan batch size yang tepat bisa membantu meningkatkan kecepatan pelatihan dan efisiensi. Dimana dalam source code menggunakan batch size 8 dalam pelatihan model.

```
EPOCHS = 10
INIT_LR = 0.001
BS = 8
# BS = 32
# opt = SGD(lr=INIT_LR, decay=0.0001)
# opt = Adam(learning_rate=INIT_LR, decay=INIT_LR / EPOCHS)
opt = Adam(learning_rate=INIT_LR)

mc = ModelCheckpoint(r'/content/Forest Fire CNN/ffs_cnn.h5',
monitor='val_accuracy', mode='max', verbose=1,
save_best_only=True)
# mc = ModelCheckpoint(r'/G:\My
Drive\Ikbal\Dataset\Training/ffs_cnn.h5',
monitor='val_accuracy', verbose=1)

default_image_size = tuple((256, 256))
image_size = 0
directory_root =
'/content/drive/MyDrive/Ikbal/Dataset/Training'
width=256
```

```
height=256
```

```
depth=3
```

3. Selanjutnya memiliki tujuan untuk membaca, mengonversi, dan menyesuaikan gambar menjadi bentuk array yang siap untuk diolah oleh model jaringan saraf tiruan.

```
def convert_image_to_array(image_dir):  
    try:  
        image = cv2.imread(image_dir)  
        if image is not None :  
            image = cv2.resize(image, default_image_size)  
            return img_to_array(image)  
        else :  
            return np.array([])  
    except Exception as e:  
        print(f"Error : {e}")  
        return None
```

4. Pada tahap kode ini melakukan beberapa hal terkait dengan memuat data gambar dari direktori tertentu dan mempersiapkannya untuk pelatihan model. Mari kita bahas langkah-langkahnya:

- a. Image list, label list = [], [] : Ini adalah inisialisasi dua list kosong yang akan digunakan untuk menyimpan array gambar dan label yang berkaitan.
- b. Root dir = listdir(directory root) : Ini mendapatkan daftar file dan folder dalam directory_root, yang merupakan lokasi utama tempat gambar-gambar dataset ditempatkan.
- c. Loop pertama (for directory in root dir): Loop ini digunakan untuk mengakses setiap folder yang mewakili kategori gambar, seperti berbagai jenis penyakit tanaman.
- d. Loop kedua (for plant disease folder in plant disease folder list): Loop ini digunakan untuk mengakses setiap folder yang mewakili kategori gambar, seperti berbagai jenis penyakit tanaman.

- e. Loop ketiga (for image in plant_disease_image_list:): Loop ini digunakan untuk mengakses setiap gambar dalam subfolder penyakit tanaman. Ini adalah bagian yang memuat dan memproses gambar.
- f. Image_directory=f"{directory_root}/{plant_folder}/{plant_disease_folder}/{image}": Ini adalah jalur lengkap ke gambar yang sedang diproses.
- g. Kondisi if image_directory.endswith(".jpg") == True or image_directory.endswith(".JPG") == True or image_directory.endswith(".jpeg") == True: Kondisi ini memastikan bahwa hanya berkas gambar dengan ekstensi .jpg, .JPG, atau .jpeg yang akan diproses.
- h. Image_list.append(convert_image_to_array(image_directory)): Gambar yang memenuhi kondisi akan diubah menjadi array dan ditambahkan ke image_list.
- i. Label_list.append(plant_disease_folder): Label yang sesuai dengan jenis penyakit dari subfolder akan ditambahkan ke label_list.

Akhir dari semua loop menunjukkan bahwa proses memuat gambar dan label selesai. Kegunaan dari kode ini adalah untuk memuat gambar-gambar dari berbagai kategori (misalnya, jenis penyakit tanaman) dalam dataset, mengubahnya menjadi bentuk array yang sesuai dengan format input yang dibutuhkan oleh model neural network, dan menghubungkannya dengan label yang sesuai. Hal ini diperlukan dalam tahap persiapan data sebelum memulai pelatihan model.

```

image_list, label_list = [], []
# try:
print("[INFO] Loading images ...")
root_dir = listdir(directory_root)
# root_dir = directory_root

for directory in root_dir :
    # remove .DS_Store from list
    if directory == ".DS_Store" :
        root_dir.remove(directory)

for plant_folder in root_dir :
    plant_disease_folder_list =
listdir(f"{directory_root}/{plant_folder}")

```

```

for disease_folder in plant_disease_folder_list :
    # remove .DS_Store from list
    if disease_folder == ".DS_Store" :
        plant_disease_folder_list.remove(disease_folder)

for plant_disease_folder in plant_disease_folder_list:
    print(f"[INFO] Processing {plant_disease_folder} ...")
    plant_disease_image_list =
listdir(f"{directory_root}/{plant_folder}/{plant_disease_folder
}/")

    for single_plant_disease_image in
plant_disease_image_list :
        if single_plant_disease_image == ".DS_Store" :
            plant_disease_image_list.remove(single_plant_di
sease_image)

        for image in plant_disease_image_list:
            image_directory =
f"{directory_root}/{plant_folder}/{plant_disease_folder}/{image
}"

            if image_directory.endswith(".jpg") == True or
image_directory.endswith(".JPG") == True or
image_directory.endswith(".jpeg") == True:
                image_list.append(convert_image_to_array(image_
directory))
                label_list.append(plant_disease_folder)

print("[INFO] Image loading completed")
# except Exception as e:
#     print(f"Error : {e}")

```

5. Tahap ini memiliki tujuan untuk menghitung jumlah total gambar yang telah dimuat ke dalam `image_list` dan kemudian menyimpan jumlah tersebut dalam variabel `image_size`. Dan juga, langkah ini adalah untuk mendapatkan informasi tentang berapa banyak data gambar yang telah dimuat dan siap untuk diproses oleh model *neural network*.

```
image_size = len(image_list)
image_size
```

6. Tahap ini berkaitan dengan pemrosesan label yang berkaitan dengan kategori (kelas) gambar yang dimuat. Dan juga, dari kode ini adalah untuk mengonversi label kategori menjadi bentuk biner yang sesuai untuk proses pelatihan, dan juga untuk menyimpan objek LabelBinarizer yang digunakan dalam format pustaka pickle. Mari kita bahas langkah-langkahnya:

- a. `label_binarizer = LabelBinarizer()`: Ini adalah objek LabelBinarizer yang digunakan untuk mengonversi label kategori menjadi bentuk biner. Objek ini digunakan untuk mentransformasi label kategori menjadi bentuk yang dapat diolah oleh model, khususnya dalam kasus klasifikasi dengan beberapa kelas.
- b. `image_labels = label_binarizer.fit_transform(label_list)`: Di sini, objek LabelBinarizer yang telah dibuat digunakan untuk mengonversi label_list (daftar label kategori) menjadi bentuk biner. Setiap kategori akan diwakili dalam bentuk biner, di mana setiap kelas akan memiliki nilai 0 atau 1 untuk setiap sampel gambar.
- c. `Pickle.dump(label_binarizer,open(r'forest_fire_smoke_transform.pkl', 'wb'))`: Ini adalah langkah yang sangat penting. Objek label_binarizer disimpan dalam format biner menggunakan modul pickle. Ini memungkinkan untuk menggunakan transformasi yang sama pada data uji atau data baru yang Anda ingin klasifikasikan di masa mendatang.
- d. `n_classes = len(label_binarizer.classes_)`: Ini menghitung jumlah total kelas atau kategori dalam dataset. Hal ini digunakan untuk menentukan jumlah unit output pada lapisan akhir dari model .

```
label_binarizer = LabelBinarizer()
image_labels = label_binarizer.fit_transform(label_list)
pickle.dump(label_binarizer,open(r'forest_fire_smoke_transform.pkl', 'wb'))
n_classes = len(label_binarizer.classes_)
```

7. Kemudian dalam tahap ini digunakan untuk mencetak daftar kelas atau kategori yang telah diidentifikasi oleh objek LabelBinarizer. Setiap kelas akan dicetak dalam urutan yang sesuai.

```
print(label_binarizer.classes_)
```

8. Dalam tahap ini memiliki tujuan untuk mengubah dan menormalkan array gambar yang ada dalam image_list. Jadi, baris ini secara efektif mengubah semua array gambar dalam image_list menjadi array float16 yang dinormalisasi antara 0 dan 1.

```
image_list = np.array(image_list, dtype=np.float16) /  
225.0
```

9. Selanjutnya tahap ini digunakan untuk membagi data gambar dan label menjadi set data latih dan data uji. dari baris ini adalah untuk membagi dataset gambar dan label menjadi dua set yang terpisah yaitu satu untuk pelatihan dan satu untuk pengujian (validasi).

```
x_train, x_test, y_train, y_test =  
train_test_split(image_list, image_labels, test_size=0.1,  
random_state=42)
```

10. Pada tahap ini digunakan untuk mengimpor modul "*garbage collection*" (gc) yang ada dalam Python. Garbage collection adalah proses otomatis pemulihan memori yang tidak terpakai agar bisa digunakan kembali.

```
import gc  
gc.collect()
```

11. Kemudian dari tahap ini digunakan untuk menghapus variabel image_list dan image_labels dari memori. Ini adalah tindakan yang disebut "penghapusan variabel" atau "dealokasi variabel". Langkah ini juga untuk membebaskan memori yang ditempati oleh variabel-variabel ini setelah selesai menggunakannya.

```
del(image_list)  
del(image_labels)
```

12. Tahap ini digunakan untuk menghasilkan variasi dari dataset gambar yang ada dengan melakukan transformasi pada gambar-gambar tersebut. Dan juga, dari kode ini adalah untuk membuat objek ImageDataGenerator

dengan konfigurasi transformasi gambar yang dapat digunakan selama pelatihan model.

```
aug = ImageDataGenerator(
    rotation_range=25, width_shift_range=0.1,
    height_shift_range=0.1, shear_range=0.2,
    zoom_range=0.2, horizontal_flip=True,
    fill_mode="nearest")
```

13. Tahap ini digunakan untuk memanggil `gc.collect()` adalah perintah yang digunakan untuk secara manual memicu proses pemulihan memori yang tidak terpakai oleh kolektor sampah (garbage collector) dalam Python. Kegunaan dari `gc.collect()` adalah untuk membebaskan memori yang tidak terpakai segera, alih-alih hanya mengandalkan garbage collector untuk menjalankan proses pemulihan memori ketika diperlukan

```
gc.collect()
```

D. Model Convolutional Neural Network (CNN)

Convolutional Neural Network adalah metode jaringan saraf mendalam, yang paling umum diterapkan untuk menganalisis gambar visual. Convolutional Neural Network juga disebut sebagai multilayer perceptron yang setiap neuronnya terhubung ke semua neuron dilapisan berikutnya. Pada tahap ini yaitu pembuatan model Convolutional Neural Network pada sistem yang akan dibuat.

```
def fit_model(n_nodes, idx):
    #####
    #define model arch

    model = Sequential()
    inputShape = (height, width, depth)
    chanDim = -1
    if K.image_data_format() == "channels_first":
        inputShape = (depth, height, width)
        chanDim = 1

    ### 1
    model.add(Conv2D(n_nodes[idx][0], (3, 3),
```

```

padding="same",input_shape=inputShape))
    model.add(Activation("relu"))
    # model.add(BatchNormalization(axis=chanDim))
    model.add(MaxPooling2D(pool_size=(3, 3)))
    # model.add(Dropout(0.25))

    ### 2
    model.add(Conv2D(n_nodes[idx][1], (3, 3), padding="same"))
    model.add(Activation("relu"))
    # model.add(BatchNormalization(axis=chanDim))
    model.add(MaxPooling2D(pool_size=(2, 2)))
    # model.add(Dropout(0.25))

    ### 3
    model.add(Conv2D(n_nodes[idx][2], (3, 3), padding="same"))
    model.add(Activation("relu"))
    # model.add(BatchNormalization(axis=chanDim))
    model.add(MaxPooling2D(pool_size=(2, 2)))
    # model.add(Dropout(0.25))

#     ## 4
#     model.add(Conv2D(n_nodes[idx][3], (3, 3),
padding="same"))
#     model.add(Activation("relu"))
#     model.add(BatchNormalization(axis=chanDim))
#     model.add(MaxPooling2D(pool_size=(2, 2)))
# #     model.add(Dropout(0.25))

model.add(Flatten())
model.add(Dense(n_nodes[idx][3]))
model.add(Activation("relu"))
# model.add(BatchNormalization())
# model.add(Dropout(0.5))
model.add(Dense(3))
model.add(Activation("softmax"))
#model.add(Activation("sigmoid"))

```

```

#####
#compile model & fit

# model.compile(loss="binary_crossentropy",
optimizer=opt,metrics=["accuracy"])
# model.compile(loss="binary_crossentropy",
metrics=["accuracy"])
model.compile(loss="categorical_crossentropy",
metrics=["accuracy"])

print(f'Training {idx+1}st Arch!')
history = model.fit(
    aug.flow(x_train, y_train, batch_size=BS),
    validation_data=(x_test, y_test),
    steps_per_epoch=len(x_train) // BS,
    epochs=EPOCHS, verbose=1, callbacks=[mc]
    # epochs=EPOCHS, verbose=1
)

#####
#plot learning curves

acc = history.history['accuracy']
epoch = range(1, len(acc) + 1)

plt.subplot(len(n_nodes), 2, (idx+1)*2-1)
plt.plot(epoch, history.history['accuracy'], label='train
acc')
plt.plot(epoch, history.history['val_accuracy'], label='val
acc')
plt.title(f'Accuracy, node {n_nodes[idx]}')
plt.legend()

plt.subplot(len(n_nodes), 2, (idx+1)*2)
plt.plot(epoch, history.history['loss'], label='train

```

```

loss')
plt.plot(epoch, history.history['val_loss'], label='val
loss')
plt.title(f'Loss, node {n_nodes[idx]}')
plt.legend()

```

Pada tahap ini kegunaan dari fungsi ini adalah untuk merancang, mengkompilasi, melatih, dan memvisualisasikan hasil dari beberapa arsitektur model yang berbeda (ditentukan oleh `n_nodes`). Fungsi ini memungkinkan untuk menguji berbagai konfigurasi arsitektur dan melihat bagaimana model bereaksi terhadap data pelatihan dan validasi.

Fungsi `fit_model (n_nodes, idx)` digunakan untuk mengatur, mengkompilasi, dan melatih model jaringan saraf tiruan (neural network) dengan arsitektur yang telah ditentukan. Mari kita bahas langkah-langkah yang dilakukan oleh fungsi ini:

1. Definisi Model: Fungsi ini memulai dengan mendefinisikan arsitektur model menggunakan modul `Sequential` dari TensorFlow Keras. Setiap layer model, termasuk layer konvolusi, aktivasi, max pooling, dan fully connected, ditambahkan sesuai dengan arsitektur yang telah ditentukan.
2. Compile Model: Setelah definisi model selesai, model dikompilasi dengan fungsi kerugian (loss function) `'binary_crossentropy'` dan metrik evaluasi `'accuracy'`. Ini adalah langkah penting sebelum model dapat dilatih.
3. Pelatihan Model: Model dilatih menggunakan fungsi `fit`. Data latih diproses dengan augmentasi yang ditentukan oleh objek `ImageDataGenerator` `aug` dan validasi dilakukan pada data uji `x_test` dan `y_test`. Epochs, ukuran batch (BS), dan parameter lainnya ditentukan.
4. Visualisasi Learning Curves: Setelah pelatihan selesai, nilai akurasi dan kerugian selama pelatihan dan validasi disimpan dalam variabel `history`. Fungsi ini mencetak kurva pembelajaran (learning curves) dalam bentuk grafik menggunakan `matplotlib`.

Fungsi ini menerima dua argumen:

1. `n_nodes`: Ini adalah daftar (list) yang berisi tiga angka yang mewakili jumlah node (filter) pada setiap lapisan konvolusi dalam arsitektur model.
2. `idx`: Ini adalah indeks yang mengacu pada urutan iterasi saat memproses arsitektur dari `n_nodes`.

Dalam pemakaian fungsi sigmoid pada saat pemodelan sangat membantu untuk meningkatkan akurasi dikarenakan fungsi sigmoid menghasilkan probabilitas output yang berkisar antara 0 dan 1. Dan itu menyebabkan tingkat akurasi meningkat dengan stabil dalam kurva akurasi.

E. Pengujian Metode Convolutional Neural Network (CNN)

Dan pengujian yang dipakai yaitu pengujian faktual dimana pengujian faktual adalah tahap dimana pengelola citra di uji menggunakan data yang sebenarnya. Tujuannya untuk memverifikasi keakuratan dan kinerja model dalam memberikan solusi dalam penyelesaian masalah dengan metode Convolutional Neural Network .

Pada tahap ini yaitu pengujian metode Convolutional Neural Network untuk menampilkan tingkat akurasi dan gambar tinggi pada tanaman jagung.

1. Pengujian Tingkat Akurasi

```
n_nodes = [
    [256, 128, 64, 32]
]

plt.figure(figsize=(13, 4*len(n_nodes)))
for x in range(len(n_nodes)):

    fit_model(n_nodes, x)
    print(f'{x+1}st Arch finished !')

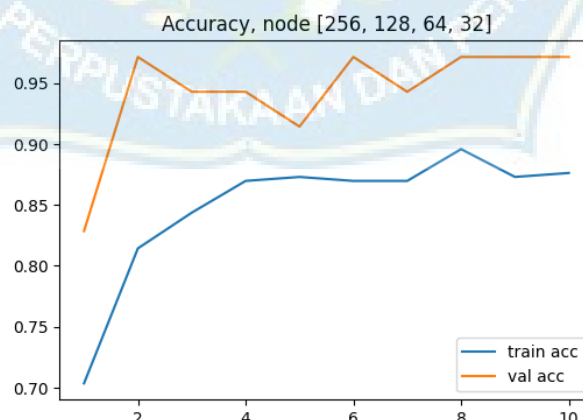
# show learning curves
plt.show()
```

Penggunaan variabel `n_nodes` untuk menguji berbagai arsitektur jaringan saraf tiruan dengan jumlah node (filter) yang berbeda pada setiap lapisan konvolusi.

Mari kita bahas langkah-langkahnya:

- a. `n_nodes`: Ini adalah variabel yang berisi daftar arsitektur yang ingin diuji. Setiap sublist dalam `n_nodes` menunjukkan jumlah filter yang akan digunakan dalam lapisan konvolusi pada setiap arsitektur.
- b. `plt.figure(figsize=(13, 4*len(n_nodes)))` : Ini adalah perintah untuk membuat gambar plot dengan ukuran yang diatur sesuai dengan banyaknya arsitektur yang ingin diuji. Setiap arsitektur akan memiliki subplot dalam gambar.
- c. Looping melalui setiap arsitektur dalam `n_nodes`:
 - 1) Memanggil fungsi `fit_model(n_nodes, x)` untuk melatih dan menguji model dengan arsitektur yang sesuai.
 - 2) Mencetak pesan yang menunjukkan bahwa arsitektur ke-`x` telah selesai dilatih dan diuji.
- d. `plt.show()`: Ini adalah perintah untuk menampilkan semua plot hasil dari proses pelatihan dan pengujian model dengan berbagai arsitektur.

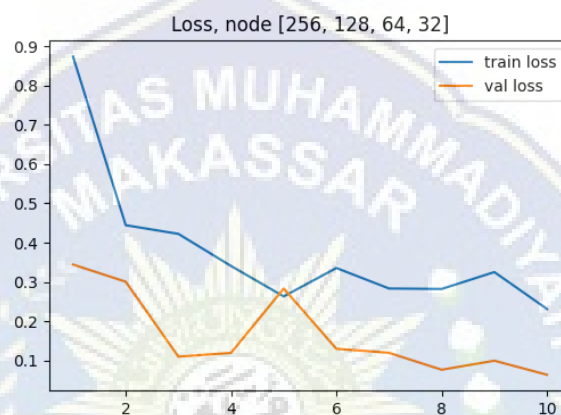
Jadi, dalam kode ini melakukan iterasi melalui daftar arsitektur dalam `n_nodes`, melatih dan menguji model menggunakan setiap arsitektur, dan akhirnya menampilkan plot hasil pelatihan dan pengujian untuk semua arsitektur yang diuji.



Gambar 12 Kurva Akurasi

Dari gambar diatas dapat dilihat bahwa kondisi kurva

akurasi meningkat dengan epoch 10 dan memperoleh nilai akurasi *training* mencapai kestabilan 80%. Pada saat epoch ke-5 sudah mencapai 0.8730 hingga epoch ke-10 dalam mendeteksi tinggi tanaman jagung. Akurasi adalah perbandingan antara data yang ingin diprediksi apakah benar sesuai dengan kelas target yang ingin dicapai dengan keseluruhan data oleh model klasifikasi. Akurasi juga digunakan untuk mengukur seberapa baik data diprediksi dengan benar.



Gambar 13 Kurva *Loss*

Dari gambar diatas dapat dilihat bahwa kondisi kurva loss menurun dengan memperoleh nilai *training loss* 0.2308 merupakan suatu fungsi dalam mengukur seberapa baik sebuah prediksi atau diagnosa dengan cara mengukur jarak hasil dari *output* prediksi dengan nilai target yang ingin dicapai. Loss juga menghasilkan nilai kesalahan atau *error* antara *output training* dan target yang ingin dicapai.

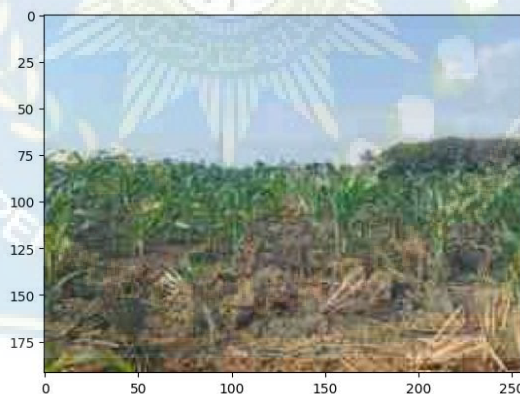
1. Pengujian Gambar Tinggi Tanaman 20 CM



Gambar 14. Tinggi Tanaman Jagung 20 cm

Dari gambar diatas dapat dilihat bahwa gambar yang terdeteksi adalah tinggi tanaman jagung dengan *title* “20 cm” yang sesuai dengan dataset *validasi*. Dan dalam pengujian identifikasi citra tinggi tanaman jagung untuk mengetahui tinggi tanaman tersebut termasuk jenis kelas tinggi tanaman yang mana.

2. Pengujian Gambar Tinggi Tanaman 50 cm



Gambar 15. Tinggi Tanaman 50 cm



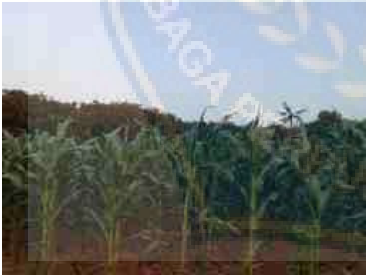
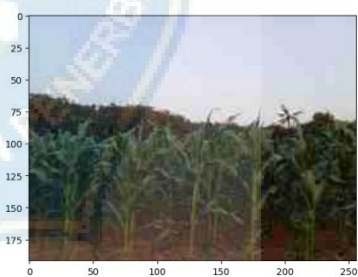

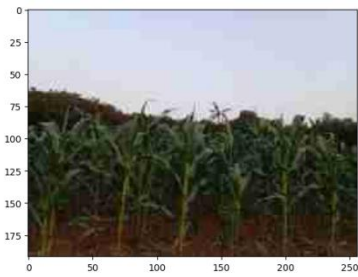
Dari gambar diatas dapat dilihat bahwa gambar yang terdeteksi adalah tinggi tanaman jagung bakteri dengan *title* “50 cm” yang sesuai dengan dataset *validasi*. Dan dalam pengujian identifikasi citra penyakit untuk mengetahui tinggi tanaman tersebut termasuk jenis kelas yang mana.

Dan dari hasil pengujian di atas mulai dari kurva akurasi dan loss, pengujian penyakit pada masing-masing kelas dengan metode *Convolutional Neural Network* (CNN) faktanya dapat membantu kinerja pengelola citra dalam mendeteksi tinggi pada tanaman jagung.

F. Hasil Pengujian Dataset Gambar Tinggi Tanaman Jagung

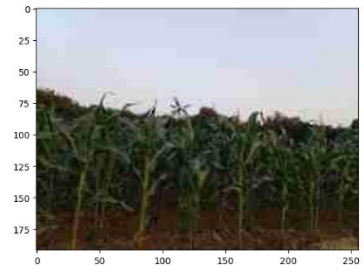
Adapun dataset gambar yang tersedia yaitu dataset gambar tinggi tanaman jagung dengan tiga kelas,. Berikut hasil diagnose dari masing- masing dataset gambar.

1.1 Table Hasil Pengujian

Tabel Hasil Pengujian		
Data Pengujian	Kelas Dataset	Hasil Deteksi
	110 CM	
	110 CM	
	110 CM	



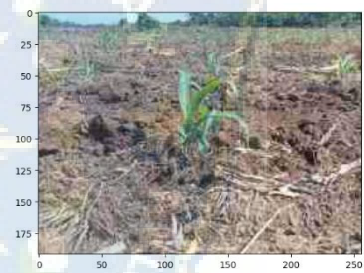
110 CM



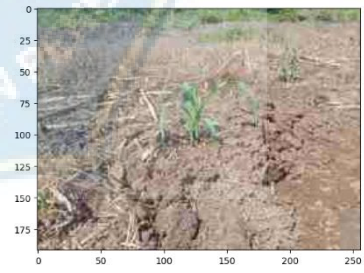
20 CM



20 CM

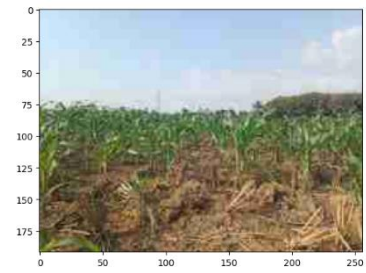


20 CM





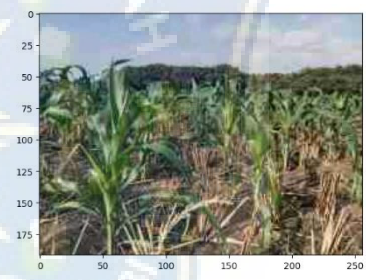
50 CM



50 CM


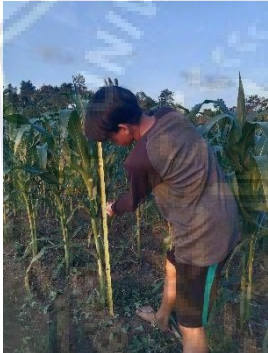



50 CM



Pada tabel diatas adalah hasil pengujian antara dataset gambar tinggi tanaman 20cm, 50cm dan 110cm pada tanaman jagung. Dimana pada dataset gambar tinggi tanaman, akurasi yang di dapat mencapai 80% untuk 5 kali percobaan testing yang dilakukan dengan hasil diagnosa adalah gambar tinggi pada tanaman jagung yaitu 20 cm, 50 cm dan 110 cm.

Tabel 1.2 Pengukuran Manual

Pengukuran	Ukuran	Akurasi
	110 CM	100%
	110 CM	100%
	50 CM	100%



50 CM

100%



20 CM

100%



20 CM

100%

Pada tabel diatas adalah hasil pengukuran manual tinggi tanaman 20cm, 50cm dan 110cm pada tanaman jagung. Dimana pada pengukuran manual tinggi tanaman, akurasi yang di dapat mencapai 100% untuk beberapa kali percobaan yang dilakukan dengan hasil diagnosa tinggi pada tanaman jagung yaitu 20 cm, 50 cm dan 110 cm.

BAB V

KESIMPULAN DAN SARAN

A. Kesimpulan

Dari hasil penelitian yang dilakukan, Adapun kesimpulan yaitu sebagai berikut:

Penelitian ini berhasil mengimplementasikan algoritma Convolutional Neural Network (CNN) untuk klasifikasi gambar tinggi tanaman jagung. CNN terbukti efektif dalam memproses dan mengenali pola visual dari citra tanaman jagung, yang dapat dihubungkan dengan tinggi tanaman secara akurat. Hasil pengujian menunjukkan bahwa model CNN yang dikembangkan mampu memberikan akurasi yang baik dalam mengklasifikasikan tinggi tanaman jagung, sehingga dapat digunakan sebagai alat bantu dalam pengelolaan pertanian, khususnya dalam pemantauan pertumbuhan tanaman.

B. Saran

Dari hasil penelitian yang dilakukan, dapat disimpulkan beberapa saran yaitu sebagai berikut:

Penelitian ini dapat dikembangkan lebih lanjut dengan menggunakan dataset yang lebih besar dan lebih beragam untuk meningkatkan akurasi dan generalisasi model CNN. Selain itu, penerapan teknik augmentasi data juga dapat dieksplorasi untuk memperkaya dataset yang ada.

DAFTAR PUSTAKA

- VBai et al. (2018). Deep Convolutional Neural Network Optimization for Extensive Image Retrieval. Pages 60–67 in Volume 303 of Neurocomputing.
- In 2017, Guo, T., Dong, J., Li, H., and Gao, Y. Convolutional Neural Network for Image Classification: A Basic Approach. IEEE Beijing.
- In 2018, Bennamoun, M.; Khan, S.; Rahmani, H.; Shah, S. A. A. Morgan & Claypool Publishers, s.l., A Guide to Convolutional Neural Networks for Computer Vision.
- Li, J., et al. (2020). "Judul Jurnal Terkait." Nama Jurnal, Volume(Issue), Halaman-Halaman. DOI: [DOI Jurnal]
- In 2019, Mohamed, O., Brahim, A., Khalid, E. A., and Mohamed, O. Convolutional Neural Networks for Content-Based Image Retrieval. pages 463–476).
- S. Naik. (2017). Fruit Grading and Classification Using Machine Vision. Volume 170 of the International Journal of Computer Application.
- In 2017, Naik, S. and Patel, B. Fruit classification based on machine vision. Journal of Computer Applications International, 170(9).
- Wang, Q., & Zhang, L. (2022). Judul Buku Terkait. Penerbit Buku. ISBN: [Nomor ISBN Buku].
- Mahdiyah, U. (2023). KLASIFIKASI KUALITAS CITRA CABAI DENGAN MENGGUNAKAN ALGORITMA *GRADIEN BOOSTING*. JAMI: Jurnal Ahli Muda Indonesia, 4(1), 61–69.
- Sely Wita, D., & Yanti Liliana, D. (2022). Klasifikasi Identitas Dengan Citra Telapak Tangan Menggunakan *Convolutional Neural Network (CNN)*. JURTI, 6(1).
- Sri Rahmadhani, U., & Lysbetti Marpaung, N. (2023). Klasifikasi Jamur Berdasarkan Genus Dengan Menggunakan Metode CNN. 8(2).
- Teknik Elektro, J., & Wega Intyanto, G. (n.d.). Klasifikasi Citra Bunga dengan Menggunakan *Deep Learning: CNN (Convolution Neural Network)*.
- Wolas, N., and M. H. Endah (undated). Convolutional Neural Network (CNN) Algorithm is Used by Deep Learning Applications to Classify Burung Familiar Accipitridae. In 2021, Telkom, U., Novamizanti, L., Rizal, S., and Zakiya, P. N. Classification of the pathology of macula retina by OCT image

usage (KLASIFIKASI PATOLOGI MAKULA RETINA MELALUI CITRA OCT USING CONVOLUTIONAL NEURAL NETWORK DENGAN, 8(5), 5072–5082).

Sun, Y., Nanekaran, Y. A., Suzauddola, M., Zhang, D., & Chen, J. (2021). Squeeze-and-excitation MobileNet model with twice transfer learning for plant disease picture identification. 15 (5) IET Image Processing, 1115–1127. The publication 10.1049/ipr2.12090

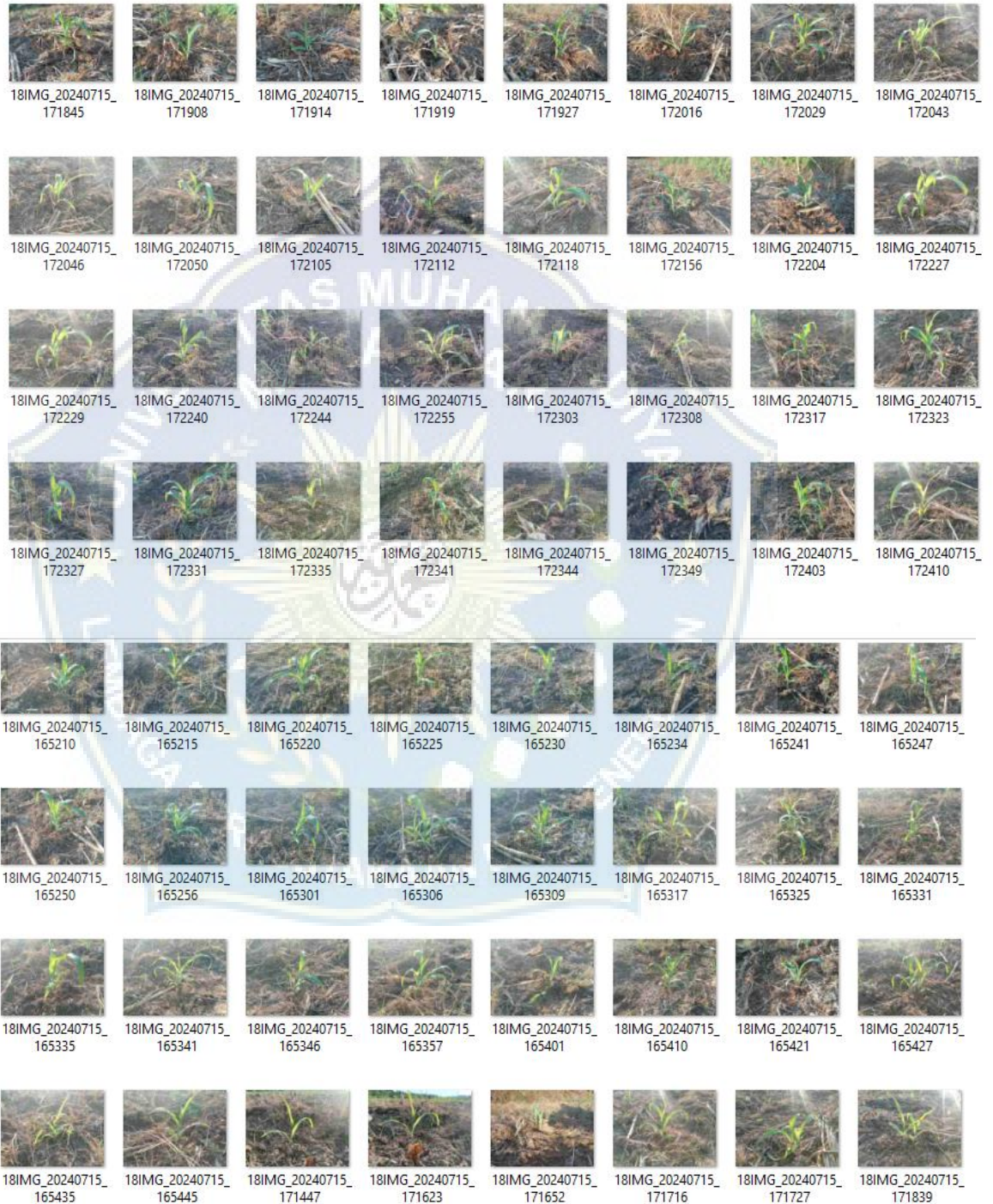
Wang, T., Warden, P., Regev, S., Rhodes, R., David, R., Duke, J., Jain, A., Reddi, V.J., Jeffries, N., Li, J., Kreeger, N., Nappier, I., and Natraj, M. (2020). Embedded Machine Learning on TinyML Systems using TensorFlow Lite Micro. 2010.08678 can be found at arxiv.org.

Susanti Diana, Yeni Astuti, and Dora Fatma Nurshanti. (2019). PRODUKSI JAGUNG MANIS DAN PENGARUH PEMBERIAN AIR TERHADAP PERTUMBUHAN (*Zea mays*). 2579–5171.



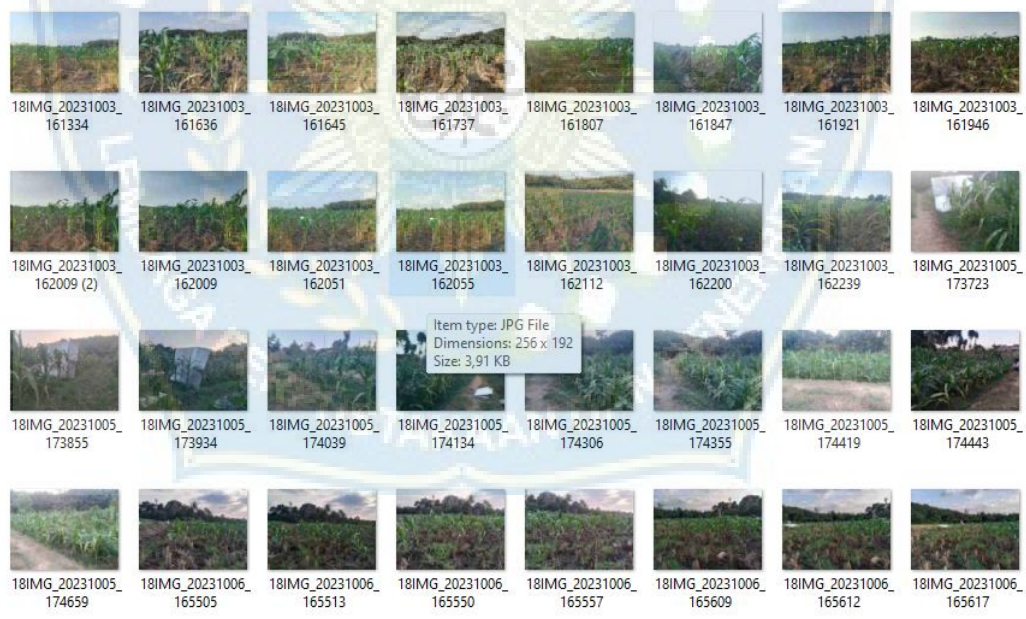
LAMPIRAN

Lampiran 1 Dataset gambar tanaman jagung ukuran 20cm





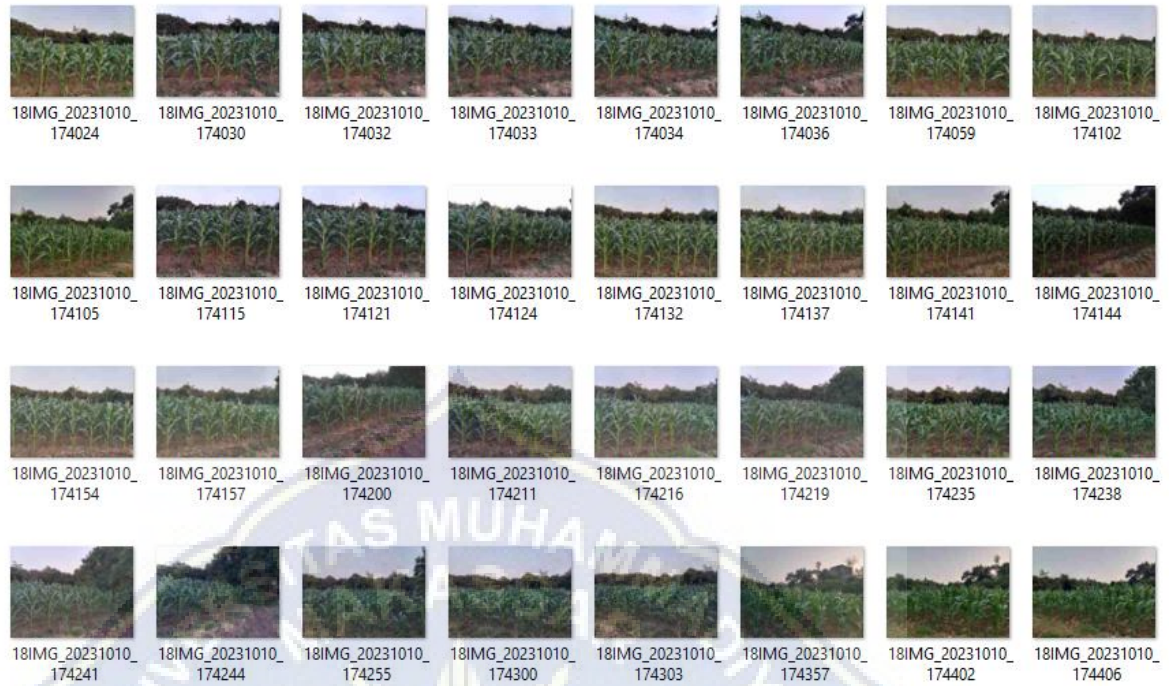
Lampiran 2 Dataset gambar tanaman jagung ukuran 50cm





Lampiran 3 Dataset gambar tanaman jagung ukuran 110cm





Lampiran 4 Source Code

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

Mounted at /content/drive

[ ] # !apt-get install unrar

[ ] # !unrar x "/G:\My Drive\Ikbal\Dataset\Training.rar" "/G:\myDrive\Trainin/"
```

```

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelBinarizer

from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import BatchNormalization
from tensorflow.keras.layers import Dense, Dropout, Flatten, Conv2D, MaxPooling2D, Activation

from tensorflow.keras import backend as K
from tensorflow.keras.preprocessing import image
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.optimizers import Adam, SGD
from tensorflow.keras.preprocessing.image import img_to_array

from tensorflow.keras.callbacks import EarlyStopping
from tensorflow.keras.callbacks import ModelCheckpoint

import numpy as np
import matplotlib.pyplot as plt
import cv2
import pickle
from os import listdir

```

```

[ ] EPOCHS = 10
    INIT_LR = 0.001
    BS = 8
    # BS = 32

    # opt = SGD(lr=INIT_LR, decay=0.0001)
    # opt = Adam(learning_rate=INIT_LR, decay=INIT_LR / EPOCHS)
    opt = Adam(learning_rate=INIT_LR)

    mc = ModelCheckpoint(r'./content/Forest Fire CNN/ffs_cnn.h5', monitor='val_accuracy', mode='max', verbose=1, save_best_only=True)
    # mc = ModelCheckpoint(r'/G:/My Drive/Ikbal/Dataset/Training/ffs_cnn.h5', monitor='val_accuracy', verbose=1)

    default_image_size = tuple((256, 256))
    image_size = 0
    directory_root = './content/drive/MyDrive/Ikbal/Dataset/Training'
    width=256
    height=256
    depth=3

```

✓ Get Size of Processed Image

```
▶ image_size = len(image_list)
image_size
```

⇄ [Show hidden output](#)

✓ Transform Image Labels using Scikit Learn's Label Binarizer

```
[ ] label_binarizer = LabelBinarizer()
image_labels = label_binarizer.fit_transform(label_list)
pickle.dump(label_binarizer,open(r'forest_fire_smoke_transform.pkl', 'wb'))
n_classes = len(label_binarizer.classes_)
```

✓ Transform Image Labels using Scikit Learn's Label Binarizer

```
[ ] label_binarizer = LabelBinarizer()
image_labels = label_binarizer.fit_transform(label_list)
pickle.dump(label_binarizer,open(r'forest_fire_smoke_transform.pkl', 'wb'))
n_classes = len(label_binarizer.classes_)
```

```
[ ] print(label_binarizer.classes_)
```

```
⇄ ['110 cm' '20 cm' '50 cm']
```

```
[ ] # np_image_list = np.array(image_list, dtype=np.float16) / 225.0
```

```
▶ image_list = np.array(image_list, dtype=np.float16) / 225.0
```

⇄ [Show hidden output](#)

✓ Splitting Data

```
[ ] x_train, x_test, y_train, y_test = train_test_split(image_list, image_labels, test_size=0.1, random_state=42)
```

Show hidden output

```
[ ] import gc
gc.collect()
```

42

```
del(image_list)
del(image_labels)
```

Show hidden output

✓ Image Generator

```
[ ] aug = ImageDataGenerator(
    rotation_range=25, width_shift_range=0.1,
    height_shift_range=0.1, shear_range=0.2,
    zoom_range=0.2, horizontal_flip=True,
    fill_mode="nearest")
```

```
[ ] gc.collect()
```

8

✓ Create Model & Training

✓ Create function to Create Model based on Model Arch

```
def fit_model(n_nodes, idx):
    #####
    #define model arch

    model = Sequential()
    inputShape = (height, width, depth)
    chanDim = -1
    if K.image_data_format() == "channels_first":
        inputShape = (depth, height, width)
        chanDim = 1

    ### 1
    model.add(Conv2D(n_nodes[idx][0], (3, 3), padding="same", input_shape=inputShape))
    model.add(Activation("relu"))
    # model.add(BatchNormalization(axis=chanDim))
    model.add(MaxPooling2D(pool_size=(3, 3)))
    # model.add(Dropout(0.25))
```



```

[ ]   ### 2
      model.add(Conv2D(n_nodes[idx][1], (3, 3), padding="same"))
      model.add(Activation("relu"))
      # model.add(BatchNormalization(axis=chanDim))
      model.add(MaxPooling2D(pool_size=(2, 2)))
      # model.add(Dropout(0.25))

      ### 3
      model.add(Conv2D(n_nodes[idx][2], (3, 3), padding="same"))
      model.add(Activation("relu"))
      # model.add(BatchNormalization(axis=chanDim))
      model.add(MaxPooling2D(pool_size=(2, 2)))
      # model.add(Dropout(0.25))

#     ## 4
#     model.add(Conv2D(n_nodes[idx][3], (3, 3), padding="same"))
#     model.add(Activation("relu"))
#     model.add(BatchNormalization(axis=chanDim))
#     model.add(MaxPooling2D(pool_size=(2, 2)))
# #     model.add(Dropout(0.25))

      model.add(Flatten())
      model.add(Dense(n_nodes[idx][3]))
      model.add(Activation("relu"))

# model.add(BatchNormalization())
# model.add(Dropout(0.5))
model.add(Dense(3))
model.add(Activation("softmax"))
#model.add(Activation("sigmoid"))

#####
#compile model & fit

# model.compile(loss="binary_crossentropy", optimizer=opt,metrics=["accuracy"])
# model.compile(loss="binary_crossentropy", metrics=["accuracy"])
model.compile(loss="categorical_crossentropy", metrics=["accuracy"])

print(f'Training {idx+1}st Arch!')
history = model.fit(
    aug.flow(x_train, y_train, batch_size=BS),
    validation_data=(x_test, y_test),
    steps_per_epoch=len(x_train) // BS,
    epochs=EPOCHS, verbose=1, callbacks=[mc]
    # epochs=EPOCHS, verbose=1
)

#####
#plot learning curves

```

```

acc = history.history['accuracy']
epoch = range(1, len(acc) + 1)

plt.subplot(len(n_nodes),2,(idx+1)*2-1)
plt.plot(epoch, history.history['accuracy'], label='train acc')
plt.plot(epoch, history.history['val_accuracy'], label='val acc')
plt.title(f'Accuracy, node {n_nodes[idx]}')
plt.legend()

plt.subplot(len(n_nodes),2,(idx+1)*2)
plt.plot(epoch, history.history['loss'], label='train loss')
plt.plot(epoch, history.history['val_loss'], label='val loss')
plt.title(f'Loss, node {n_nodes[idx]}')
plt.legend()

```

✓ Launch Train Function

```

[ ] n_nodes = [
    [256, 128, 64, 32]
]

plt.figure(figsize=(13, 4*len(n_nodes)))
for x in range(len(n_nodes)):

    fit_model(n_nodes, x)
    print(f'{x+1}st Arch finished !')

# show learning curves
plt.show()

```

```

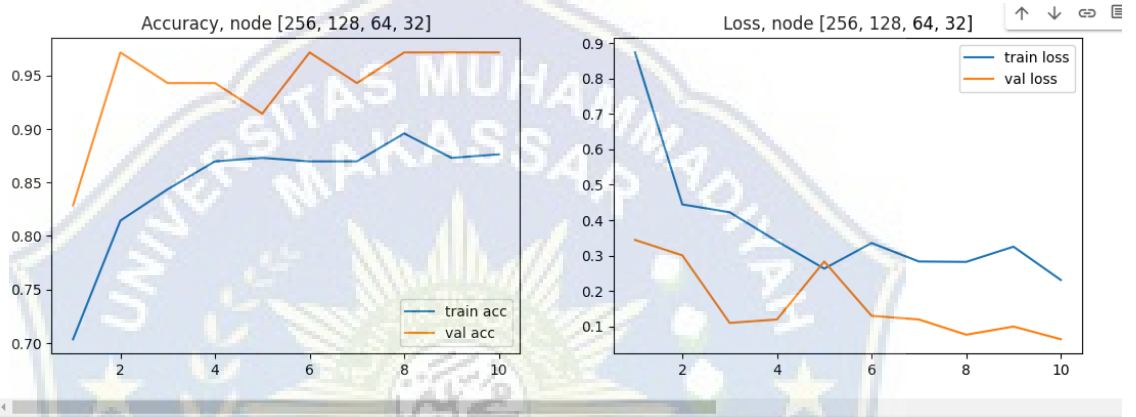
Epoch 1/10
39/39 [=====] - ETA: 0s - loss: 0.8742 - accuracy: 0.7036
Epoch 1: val_accuracy improved from -inf to 0.82857, saving model to /content/Forest Fire CNN/ffs_cnn.h5
39/39 [=====] - 198s 5s/step - loss: 0.8742 - accuracy: 0.7036 - val_loss: 0.3441 - val_accuracy: 0.8286
Epoch 2/10
/usr/local/lib/python3.10/dist-packages/keras/src/engine/training.py:3103: UserWarning: You are saving your model as an HDF5 file via `model.save()`. This file
- saving_api.save_model(
39/39 [=====] - ETA: 0s - loss: 0.4444 - accuracy: 0.8143
Epoch 2: val_accuracy improved from 0.82857 to 0.97143, saving model to /content/Forest Fire CNN/ffs_cnn.h5
39/39 [=====] - 187s 5s/step - loss: 0.4444 - accuracy: 0.8143 - val_loss: 0.3004 - val_accuracy: 0.9714
Epoch 3/10
39/39 [=====] - ETA: 0s - loss: 0.4223 - accuracy: 0.8436
Epoch 3: val_accuracy did not improve from 0.97143
39/39 [=====] - 192s 5s/step - loss: 0.4223 - accuracy: 0.8436 - val_loss: 0.1094 - val_accuracy: 0.9429
Epoch 4/10
39/39 [=====] - ETA: 0s - loss: 0.3405 - accuracy: 0.8697
Epoch 4: val_accuracy did not improve from 0.97143
39/39 [=====] - 200s 5s/step - loss: 0.3405 - accuracy: 0.8697 - val_loss: 0.1191 - val_accuracy: 0.9429
Epoch 5/10
39/39 [=====] - ETA: 0s - loss: 0.2629 - accuracy: 0.8730
Epoch 5: val_accuracy did not improve from 0.97143
39/39 [=====] - 191s 5s/step - loss: 0.2629 - accuracy: 0.8730 - val_loss: 0.2830 - val_accuracy: 0.9143
Epoch 6/10
39/39 [=====] - ETA: 0s - loss: 0.3353 - accuracy: 0.8697
Epoch 6: val_accuracy did not improve from 0.97143

```

```

Epoch 6/10
39/39 [=====] - ETA: 0s - loss: 0.3353 - accuracy: 0.8697
Epoch 6: val_accuracy did not improve from 0.97143
39/39 [=====] - 190s 5s/step - loss: 0.3353 - accuracy: 0.8697 - val_loss: 0.1295 - val_accuracy: 0.9714
Epoch 7/10
39/39 [=====] - ETA: 0s - loss: 0.2831 - accuracy: 0.8697
Epoch 7: val_accuracy did not improve from 0.97143
39/39 [=====] - 192s 5s/step - loss: 0.2831 - accuracy: 0.8697 - val_loss: 0.1191 - val_accuracy: 0.9429
Epoch 8/10
39/39 [=====] - ETA: 0s - loss: 0.2819 - accuracy: 0.8958
Epoch 8: val_accuracy did not improve from 0.97143
39/39 [=====] - 187s 5s/step - loss: 0.2819 - accuracy: 0.8958 - val_loss: 0.0760 - val_accuracy: 0.9714
Epoch 9/10
39/39 [=====] - ETA: 0s - loss: 0.3249 - accuracy: 0.8730
Epoch 9: val_accuracy did not improve from 0.97143
39/39 [=====] - 189s 5s/step - loss: 0.3249 - accuracy: 0.8730 - val_loss: 0.0991 - val_accuracy: 0.9714
Epoch 10/10
39/39 [=====] - ETA: 0s - loss: 0.2308 - accuracy: 0.8762
Epoch 10: val_accuracy did not improve from 0.97143
39/39 [=====] - 197s 5s/step - loss: 0.2308 - accuracy: 0.8762 - val_loss: 0.0632 - val_accuracy: 0.9714
1st Arch finished !

```



```

[ ] from keras.models import load_model
[ ] fs_7030 = load_model('/content/Forest Fire CNN/ffs_cnn.h5')
[ ] loss, acc = fs_7030.evaluate(x_train, y_train)
10/10 [=====] - 54s 6s/step - loss: 0.3880 - accuracy: 0.8857
[ ] loss, acc = fs_7030.evaluate(x_test, y_test)
2/2 [=====] - 5s 438ms/step - loss: 0.3004 - accuracy: 0.9714

```

Load & Testing

```

[ ] from tensorflow.keras.models import load_model
[ ] cs_adam_10 = load_model('/content/Forest Fire CNN/ffs_cnn.h5')

```

Load & Testing

```
[ ] from tensorflow.keras.models import load_model
```

```
[ ] cs_adam_10 = load_model('/content/Forest Fire CNN/ffs_cnn.h5')
```

```
[ ] gmb1 = cv2.imread('/content/drive/MyDrive/Ikbal/Dataset/Training/train/50 cm/18IMG_20231003_161636.jpg')
```

```
[ ] image_rgb = cv2.cvtColor(gmb1, cv2.COLOR_BGR2RGB)
plt.imshow(image_rgb)
```

```
<matplotlib.image.AxesImage at 0x7e2e9c1296c0>
```



```
[ ] # Prepare image test
```

```
# img_test = cv2.imread('/content/drive/MyDrive/Skripsi/rumput laut/rumput laut/R1_bulu_kucing/IMG-20230603-WA0173.jpg')
img3 = cv2.resize(gmb1, default_image_size)
img3 = img_to_array(img3)
img3 = np.expand_dims(img3, axis=0)
img3 = img3/225
```

```
img3.shape
```

```
(1, 256, 256, 3)
```

```
[ ] y_pred = cs_adam_10.predict(img3)
```

```
1/1 [=====] - 0s 264ms/step
```

```
[ ] y_pred
```

```
array([[0.02830757, 0.93724257, 0.03444988]], dtype=float32)
```

```
[ ] # for x, y in enumerate(y_pred):
```

```

[ ] y_pred
array([[0.02830757, 0.93724257, 0.03444988]], dtype=float32)

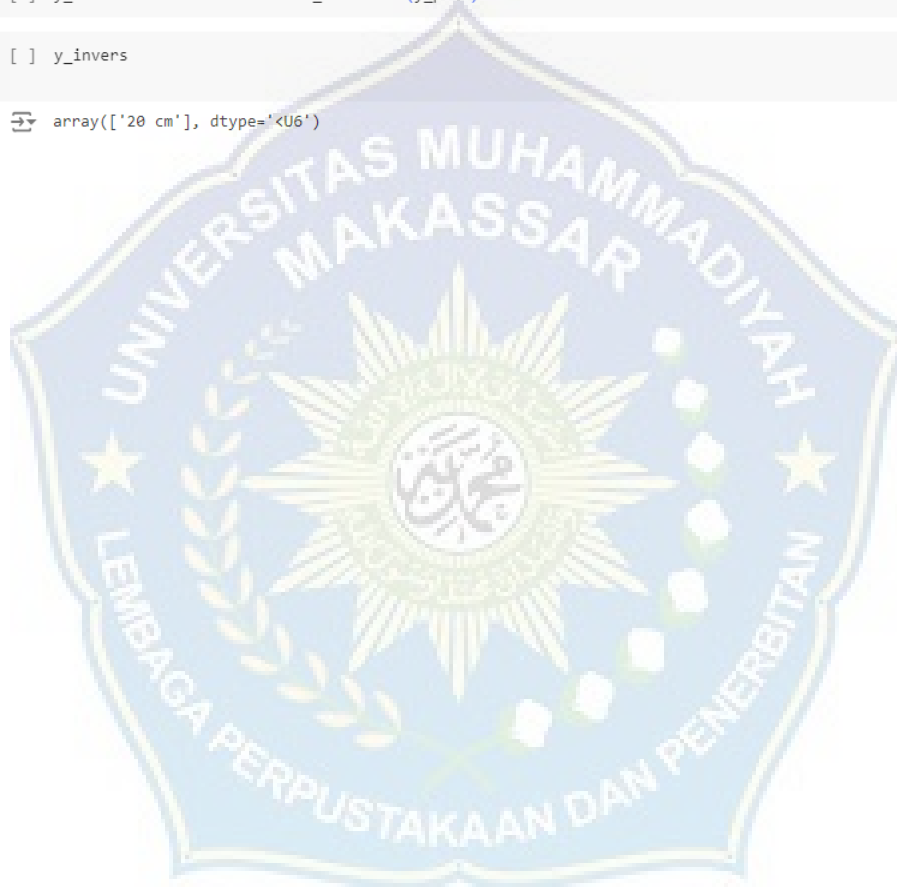
[ ] # for x, y in enumerate(y_pred):
    # positive = np.where(y_pred[x] == np.amax(y_pred[x]))
    # new = np.zeros(5, dtype=int)
    # new[positive] = 1
    # y_pred[x] = new

[ ] with open('/content/forest_fire_smoke_transform.pkl', 'rb') as binarizer_file:
    binarizer = pickle.load(binarizer_file)

[ ] y_invers = binarizer.inverse_transform(y_pred)

[ ] y_invers
array(['20 cm'], dtype='<U6')

```



Lampiran 5 Surat

**UNIVERSITAS MUHAMMADIYAH MAKASSAR**
FAKULTAS TEKNIK
PROGRAM STUDI INFORMATIKA

بِسْمِ اللّٰهِ الرَّحْمٰنِ الرَّحِیْمِ

Nomor : 372/05/C.4-VI/V45/2024 Makassar, 21 Dzulqaidah 1445 H
Lamp - 29 Mei 2024 M
Hal Pengantar Penelitian

Kepada yang Terhormat,
Ketun LP3M Unismuh Makassar
Di -
Tempat

Assalamu 'Alaikum Warahmatullahi Wabarakatuh

Dengan Rahmat Allah SWT, Semoga aktivitas kita bernilai ibadah di Sisi - Nya Dalam tanyaka penyelesaian Tugas Sarjana / Tugas Akhir Mahasiswa pada Program Studi Informatika Fakultas Teknik Universitas Muhammadiyah Makassar dengan judul "Pengukuran Tinggi Tanaman Berdasarkan klasifikasi Image Dengan Menggunakan Algoritma CNN", Sehubungan hal tersebut maka kami meminta kesediaan Bapak/Ibu agar kiranya berkenan membantu perihal surat tersebut Bersama ini kami sampaikan mahasiswa(i)

No.	Stambuk	Nama
1	105 84 11036 20	Ikhbal

Demikian surat kami atas perhatian dan kerja samanya kami haturkan banyak terima kasih
Jazahumullahi Khairan Katsiran
Wassalamu 'Alaikum warahmatullahi Wabarakatuh


Ketun Program Studi
A. M. Hayat, S.Kom., M.T.,
NIP. 196301011980001000

Tembusan: Kepada Yang Terhormat,
1. Dekan Fakultas Teknik
2. Arsip

LEMBAGA PERUSAHAAN DAN PENERBITAN

Gedung Mesra Iqbal Lantai 1
Jl. Sultan Alauddin No. 259 Tello, (0411) 366 872 Fax (0411) 065 580 Makassar 90221
Web: <http://stakam.uin-makassar.ac.id> e-mail: stakam@uin-makassar.ac.id





MAJELIS PENDIDIKAN TINGGI PIMPINAN PUSAT MUHAMMADIYAH
UNIVERSITAS MUHAMMADIYAH MAKASSAR

LEMBAGA PENELITIAN PENGEMBANGAN DAN PENGABDIAN KEPADA MASYARAKAT

Jl. Sultan Alauddin No. 259 Telp. 085972 Fax (0811)865581 Makassar 90221 e-mail: lp3m@unmah.ac.id

Nomor : 4389/05/C.4-VIII/V/1445/2024

30 May 2024 M

Lamp : 1 (satu) Rangkap Proposal

22 Dzulqa'dah 1445

Hal : Permohonan Izin Penelitian

Kepada Yth,

Bapak Gubernur Prov. Sul-Sel

Cq. Kepala Dinas Penanaman Modal & PTSP Provinsi Sulawesi Selatan

di -

Makassar

بِسْمِ اللّٰهِ الرَّحْمٰنِ الرَّحِیْمِ

Berdasarkan surat Dekan TEKNIK Universitas Muhammadiyah Makassar, nomor: 372/05/C.4-VI/V/45/2024 tanggal 29 Mei 2024, menerangkan bahwa mahasiswa tersebut di bawah ini :

Nama : **IKBAL**

No. Stambuk : **10584 1103-620**

Fakultas : **TEKNIK**

Jurusan : **INFORMATIKA**

Pekerjaan : **Mahasiswa**

Bermaksud melaksanakan penelitian/pengumpulan data dalam rangka penulisan Skripsi dengan judul :

"Pengaruh Tinggi Tanaman Berdasarkan Klasifikasi Image dengan Menggunakan Algoritma CNN"

Yang akan dilaksanakan dari tanggal 31 Mei 2024 s/d 31 Juli 2024.

Sehubungan dengan maksud di atas, kiranya Mahasiswa tersebut diberikan izin untuk melakukan penelitian sesuai ketentuan yang berlaku.

Demikian, atas perhatian dan kerjasamanya diucapkan Jazakumullahu khaeran

بِسْمِ اللّٰهِ الرَّحْمٰنِ الرَّحِیْمِ

Ketua LP3M,



Dr. Muh. Arief Muhsin, M.Pd.

NBM 1127761

05-24



PEMERINTAH PROVINSI SULAWESI SELATAN
DINAS PENANAMAN MODAL DAN PELAYANAN TERPADU SATU PINTU

Jl. Bougainville No.5 Telp. (0411) 441077 Fax. (0411) 448936
Website : <http://simap-new.sulselprov.go.id> Email : ptsp@sulselprov.go.id
Makassar 90231

Nomor : 13920/S.01/PTSP/2024 Kepada Yth.
Lampiran : - Bupati Gowa
Perihal : izin penelitian

di-
Tempat

Berdasarkan surat Ketua LP3M UNISMUH Makassar Nomor : 4389/05/C.4-VIII/V/1445/2024 tanggal 30 Mei 2024 perihal tersebut diatas, mahasiswa/peneliti dibawah ini:

Nama : IKBAL
Nomor Pokok : 105841103620
Program Studi : Teknik Informatika
Pekerjaan/Lembaga : Mahasiswa (D4)
Alamat : Jl. Sultan Alauddin No. 259 Makassar

PROVINSI SULAWESI SELATAN

Bermaksud untuk melakukan penelitian di daerah/kantor saudara dalam rangka menyusun KARYA TULIS, dengan judul :

" KLASIFIKASI IMAGE TINGGI TANAMAN JAGUNG DENGAN MENGGUNAKAN ALGORITMA CONVOLUTION NEURAL NETWORK (CNN) "

Yang akan dilaksanakan dari : Tgl. 30 Mei s/d 30 Juni 2024

Sehubungan dengan hal tersebut diatas, pada prinsipnya kami *menyetujui* kegiatan dimaksud dengan ketentuan yang tertera di belakang surat izin penelitian.

Demikian Surat Keterangan ini diberikan agar dipergunakan sebagaimana mestinya.

Diterbitkan di Makassar
Pada Tanggal 30 Mei 2024

KEPALA DINAS PENANAMAN MODAL DAN PELAYANAN TERPADU
SATU PINTU PROVINSI SULAWESI SELATAN



ASRUL SANI, S.H., M.Si.
Pangkat : PEMBINA TINGKAT I
Nip : 19750321 200312 1 008

Tembusan Yth
1. Ketua LP3M UNISMUH Makassar di Makassar;
2. Peringgal



**MAJELIS PENDIDIKAN TINGGI PIMPINAN PUSAT MUHAMMADIYAH
UNIVERSITAS MUHAMMADIYAH MAKASSAR
UPT PERPUSTAKAAN DAN PENERBITAN**

Alamat kantor: Jl.Sultan Alauddin NO.259 Makassar 90221 Tlp.(0411) 866972,881593, Fax.(0411) 865588

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

SURAT KETERANGAN BEBAS PLAGIAT

UPT Perpustakaan dan Penerbitan Universitas Muhammadiyah Makassar,
Menerangkan bahwa mahasiswa yang tersebut namanya di bawah ini:

Nama : Iqbal

Nim : 105841103620

Program Studi : Teknik Informatika

Dengan nilai:

No	Bab	Nilai	Ambang Batas
1	Bab 1	9 %	10 %
2	Bab 2	24 %	25 %
3	Bab 3	0 %	15 %
4	Bab 4	10 %	10 %
5	Bab 5	5 %	5 %

Dinyatakan telah lulus cek plagiat yang diadakan oleh UPT- Perpustakaan dan Penerbitan Universitas Muhammadiyah Makassar Menggunakan Aplikasi Turnitin.

Demikian surat keterangan ini diberikan kepada yang bersangkutan untuk dipergunakan seperlunya.

Makassar, 16 Agustus 2024

Mengetahui,

Kepala UPT- Perpustakaan dan Penerbitan,



Nursilikh, S.Hum.,M.I.P
NBM. 964 591

Ikbal 105841103620 BAB I

by Tahap Tutup



Submission date: 15-Aug-2024 11:43AM (UTC+0700)

Submission ID: 2432288750

File name: BAB1_3.docx (20.94K)

Word count: 749

Character count: 5174

Ikbal 105841103620 BAB I

ORIGINALITY REPORT

9%

SIMILARITY INDEX

6%

INTERNET SOURCES

3%

PUBLICATIONS

%

STUDENT PAPERS

PRIMARY SOURCES

- 1 Fajar Sudana Putra, Kusri, Mei P Kurniawan. "Deteksi Otomatis Jerawat Wajah Menggunakan Metode Convolutional Neural Network (CNN)", Journal of Information Technology, 2021
Publication 3%
- 2 makalah3satu.blogspot.com
Internet Source 3%
- 3 docplayer.info
Internet Source 2%
- 4 text-id.123dok.com
Internet Source 2%

Exclude quotes On

Exclude matches < 2%

Exclude bibliography On

Ikbal 105841103620 BAB II

by Tahap Tutup



Submission date: 15-Aug-2024 11:44AM (UTC+0700)

Submission ID: 2432289106

File name: BAB2_2.docx (226.24K)

Word count: 1846

Character count: 12156

Ikbal 105841103620 BAB II

ORIGINALITY REPORT

24%	21%	9%	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	digilibadmin.unismuh.ac.id Internet Source	9%
2	jurnal.unej.ac.id Internet Source	3%
3	journal.akb.ac.id Internet Source	2%
4	fikes.respati.ac.id Internet Source	2%
5	media.neliti.com Internet Source	2%
6	impiclub.com Internet Source	1%
7	download.garuda.ristekdikti.go.id Internet Source	1%
8	www.e-jmii.org Internet Source	1%
9	Diesti Mutia Rahmi, Rinda Yanti. "PEMANFAATAN KOMPOS LIMBAH PASAR UNTUK MENINGKATKAN PERTUMBUHAN	1%



DAN PRODUKSI JAGUNG (*Zea mays* L.)",
Journal of Food Crop and Applied Agriculture,
2020
Publication

10 Alang Mulya Lesmana, Ronna Putri Fadhillah,
Chaerur Rozikin. "Identifikasi Penyakit pada
Citra Daun Kentang Menggunakan
Convolutional Neural Network (CNN)", Jurnal
Sains dan Informatika, 2022
Publication

11 medium.com
Internet Source 1 %

12 Naflah Faulina. "Implementation Of Artificial
Neural Network (ANN) Classification In Type 2
Diabetes Mellitus Cases", Sciencestatistics:
Journal of Statistics, Probability, and Its
Application, 2024
Publication

13 resepimasakan.net
Internet Source <1 %

14 basomadiong.wordpress.com
Internet Source <1 %

15 Umi Mahdiah. "KLASIFIKASI KUALITAS CITRA
CABAI DENGAN MENGGUNAKAN ALGORITMA
GRADIEN BOOSTING", JAMI: Jurnal Ahli Muda
Indonesia, 2023
Publication

16 Aris Purwanto, Siti Nurul Afiyah. "Sistem Peramalan Produksi Jagung Provinsi Jawa Barat Menggunakan Metode Double Exponential Smoothing", Jurnal Ilmiah Teknologi Informasi Asia, 2020
Publication <1 %

17 Errissya Rasywir, Rudolf Sinaga, Yovi Pratama. "Analisis dan Implementasi Diagnosis Penyakit Sawit dengan Metode Convolutional Neural Network (CNN)", Paradigma - Jurnal Komputer dan Informatika, 2020
Publication <1 %

18 Norberth Wolas, Hamzah Hamzah, Marselina Endah Hiswati. "Aplikasi Deep Learning Menggunakan Algoritma Convolutional Neural Network (CNN) Untuk Klasifikasi Burung Famili Accipitridae", Seminar Nasional Teknik Elektro, Informatika dan Sistem Informasi, 2022
Publication <1 %

Exclude quotes Off Exclude matches Off
Exclude bibliography Off

Ikbal 105841103620 BAB III

by Tahap Tutup



Submission date: 15-Aug-2024 03:37PM (UTC+0700)

Submission ID: 2432355682

File name: BAB3_2.docx (21.23K)

Word count: 364

Character count: 2343

Ikbal 105841103620 BAB III

ORIGINALITY REPORT

0%

SIMILARITY INDEX

0%

INTERNET SOURCES

0%

PUBLICATIONS

%

STUDENT PAPERS

PRIMARY SOURCES



Exclude quotes

Off

Exclude matches

Off

Exclude bibliography

Off



Ikbal 105841103620 BAB IV

by Tahap Tutup



Submission date: 15-Aug-2024 03:42PM (UTC+0700)

Submission ID: 2432357024

File name: BAB4_2.docx (1.46M)

Word count: 694

Character count: 4167

Ikbal 105841103620 BAB IV

ORIGINALITY REPORT

10% SIMILARITY INDEX
9% INTERNET SOURCES
7% PUBLICATIONS
% STUDENT PAPERS

PRIMARY SOURCES

1	ejournal.unma.ac.id Internet Source		4%
2	digilibadmin.unismuh.ac.id Internet Source		2%
3	Mursyid Sinung Nugroho, Eddy Nurraharjo. "Klasifikasi Hama Tanaman Padi berdasarkan Citra Daun Menggunakan Metode Convolutional Neural Network", BIOEDUSAINS:Jurnal Pendidikan Biologi dan Sains, 2023 Publication		2%
4	repo.stis.ac.id Internet Source		2%

Exclude quotes On
Exclude bibliography On
Exclude matches < 2%

Ikbal 105841103620 BAB V

by Tahap Tutup



Submission date: 14-Aug-2024 01:36PM (UTC+0700)

Submission ID: 2431864467

File name: BAB_V_-_2024-08-14T143534.018.docx (26.99K)

Word count: 155

Character count: 1001

Ikkal 105841103620 BAB V

ORIGINALITY REPORT

5%

SIMILARITY INDEX

5%

INTERNET SOURCES

0%

PUBLICATIONS

%

STUDENT PAPERS

PRIMARY SOURCES

1

repository.uinbanten.ac.id
Internet Source

5%



Exclude quotes

On

Exclude matches

< 1%

Exclude bibliography

On

