

**PERBANDINGAN FUNGSI AKTIVASI HIDDEN LAYER DAN OUTPUT  
LAYER STUDI KASUS SISTEM PAKAR PENYAKIT AYAM**

**SKRIPSI**

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



**ISNAWATI**

**105841110319**

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

**2023**

**PERBANDINGAN FUNGSI AKTIVASI HIDDEN LAYER DAN OUTPUT  
LAYER STUDI KASUS SISTEM PAKAR PENYAKIT AYAM**

**Diajukan Untuk Memenuhi Salah Satu Syarat Guna Memperoleh  
Gelara Sarjana Komputer Prodi Informatika Fakultas Teknik  
Universitas Muhammadiyah Makassar**

**Disusun Dan Diajukan Oleh:**

**ISNAWATI  
105841110319**

**PROGRAM STUDI INFORMATIKA  
FAKULTAS TEKNIK  
UNIVERSITAS MUHAMMADIYAH MAKASSAR  
2023**



UNIVERSITAS MUHAMMADIYAH MAKASSAR

FAKULTAS TEKNIK



Kampus  
Merdeka  
INDONESIA JAYA

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](http://www.unismuh.ac.id), e\_mail: [unismuh@gmail.com](mailto:unismuh@gmail.com)

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

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

### PENGESAHAN

Skripsi atas nama Isnawati dengan nomor induk Mahasiswa 105 84 1110319, dinyatakan diterima dan disahkan oleh Panitia Ujian Tugas Akhir/Skripsi sesuai dengan Surat Keputusan Dekan Fakultas Teknik Universitas Muhammadiyah Makassar Nomor : 405/05/A.5-II/VIII/45/2023, sebagai salah satu syarat guna memperoleh gelar Sarjana Komputer pada Program Studi Informatika Fakultas Teknik Universitas Muhammadiyah Makassar pada hari Sabtu tanggal 31 Agustus 2023.

Panitia Ujian :

Makassar, 02 Safar 1444 H  
31 Agustus 2023 M

1. Pengawas Umum

a. Rektor Universitas Muhammadiyah Makassar

Prof. Dr. H. AMBO ASSE, M.Ag

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

3. Anggota : 1. Rizki Yusliana Bakti, ST., MT

2. Lukman Anas, S.Kom., MT

3. Lukman, S.Kom., M.T.

Mengetahui :

Pembimbing I


Pembimbing II

  
Titin Wahyuni, S.Pd., MT.

  
Fahrim Irhamna Rahman, S.Kom., MT.



Dekan Fakultas Teknik

  
Dr. Ir. H. Nurnawaty, ST., MT., IPM

NBM : 795 108



UNIVERSITAS MUHAMMADIYAH MAKASSAR

## FAKULTAS TEKNIK



Kampus  
Merdeka  
INDONESIA JAYA

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](http://www.unismuh.ac.id), e\_mail: [unismuh@gmail.com](mailto:unismuh@gmail.com)

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

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

### HALAMAN PERSETUJUAN

Tugas Akhir ini diajukan untuk memenuhi syarat ujian guna memperoleh gelar Sarjana Komputer (S.Kom) Program Studi Informatika Fakultas Teknik Universitas Muhammadiyah Makassar.

Judul Skripsi : **PERBANDINGAN FUNGSI AKTIVASI HIDDEN LAYER DAN OUTPUT LAYER STUDI KASUS SISTEM PAKAR PENYAKIT AYAM**

Nama : ISNAWATI

Stambuk : 105841110319

Makassar, 31 Agustus 2023

Telah Diperiksa dan Disetujui  
Oleh Dosen Pembimbing;

Pembimbing I

Titin Wahyuni, S.Pd., MT.

Pembimbing II

Fahrim Irhamna Rahman, S.Kom., MT.

Mengetahui,

Ketua Program Studi Informatika

  
Muhyidin A.M Hayat, S.kom, M.T  
PRODI INFORMATIKA NBM : -

## MOTTO DAN PERSEMBAHAN

### *Motto*

*"Sesungguhnya bersama kesulitan itu ada kemudahan, maka apabila kamu telah selesai (dari suatu urusan), tetaplah bekerja keras (untuk urusan yang lain)."*

*-QS Al-Insyirah : 6-7-*

### *Persembahan*

*Ku persembahkan skripsi ini kepada orang-orang yang sangat kusayangi. Karya ini merupakan bentuk rasa syukur saya dengan perasaan bahagia dan rasa hormat, penuh cinta dan kasih, serta terima kasih kupersembahkan dan kuhadiakan kepada kedua orang tuaku, Ayahanda Kamaruddin, S.Pd dan ibunda Hasmawati yang telah merawat aku dari kecil, membimbing dan bekerja keras untuk membayai sekolahku serta tiada henti-hentinya mendoakan untuk kesuksesanku.*



## ABSTRAK

Ayam merupakan hewan peliharaan yang adaptif dan dapat dikatakan bisa hidup di sembarang tempat, asalkan tersedianya makanan. Namun tak sedikit juga ayam yang terserang penyakit yang dapat merugikan para peternak ayam sehingga mengakibatkan kerugian. Ayam bisa terserang penyakit karena faktor lingkungan atau kondisi tubuh ayam yang bersangkutan. Secara umum, penyebab tersebut dikategorikan ke dalam tiga faktor, yaitu infeksi, perkembangan tingkat resistensi, dan penyakit yang tersembunyi, diperlukan pendeteksian penyakit ayam. Dalam kasus pendeteksian objek, salah satu metode yang bisa digunakan yaitu Convolutional Neural Network (CNN). CNN salah satu metode dari deep learning yang dapat melakukan proses pembelajaran mandiri untuk mengenali suatu objek, ekstraksi objek dan klasifikasi serta dapat diterapkan pada citra resolusi tinggi dan bekerja lebih baik dibandingkan algoritma lainnya. Penelitian ini dilakukan untuk melakukan perbandingan fungsi aktivasi mana yang lebih baik dalam mempercepat perubahan model dan akurasi. Hasil penelitian ini menunjukkan bahwa fungsi aktivasi, nilai epoch, dan batch size dapat mempengaruhi akurasi model CNN. Pada penelitian ini, fungsi aktivasi relu-softmax, jumlah epoch 150 dan nilai batch size 8 menghasilkan akurasi terbaik dengan nilai akurasi training sebesar 92% dan akurasi pengujian pada 11 citra adalah sebesar 100%. Dibandingkan dengan fungsi aktivasi softmax-relu, jumlah epoch 150 dan nilai batch size 8 menghasilkan akurasi training yang rendah sebesar 26%.

**Kata kunci:** Ayam, Fungsi Aktivasi, Sistem Pakar



## **ABSTRACT**

*Chickens are adaptive pets and can be said to be able to live in any place, as long as food is available. However, not a few chickens are also attacked by diseases that can be detrimental to chicken farmers, resulting in losses. Chickens can get sick due to environmental factors or the condition of the chicken's body. In general, these causes are categorized into three factors, namely infection, development of resistance levels, and hidden diseases, it is necessary to detect chicken diseases. In the case of object detection, one method that can be used is Convolutional Neural Network (CNN). CNN is a method of deep learning that can carry out independent learning processes to recognize an object, object extraction and classification and can be applied to high-resolution images and works better than other algorithms. This research was conducted to compare which activation function is better in accelerating model changes and accuracy. The results of this study indicate that the activation function, epoch value, and batch size can affect the accuracy of the CNN model. In this study, the activation function relu-softmax, the number of epochs is 150 and the batch size value is 8 produces the best accuracy with a training accuracy value of 92% and testing accuracy on 11 images is 100%. Compared to the softmax-relu activation function, the number of epochs of 150 and a batch size of 8 results in a low training accuracy of 26%.*

**Keywords:** Activation Function, Chicken, CNN, Expert System



## KATA PENGANTAR

الرَّحِيمِ الرَّحْمَنِ اللَّهُ بِسْمِ

Segala puji bagi Allah SWT yang telah membeikan rahmat dan hidayahnya kepada penlis, sholawat beserta salam semoga tercurahkan kepada Nabi Muhammad SAW sang revolusioner sejati yang menjadi suri tauladan seluruh ummat, yang telah menyebarkan islam hingga sampai detik ini kita masih merasakan nikmat berislam. Sehingga penulis dapat menyelesaikan skripsi dengan judul "Perbandingan Fungsi Aktivasi Hidden Layer Dan Output Layer Studi Kasus Sistem Pakar Penyakit Ayam".

Skripsi ini disusun oleh penulis guna memenuhi salah satu syarat untuk menyelesaikan Program Sarjana (S1) Pada Program Studi Informatika Fakultas Teknik Universitas Muhammadiyah Makassar. Penulis berharap, dengan adanya skripsi ini dapat menambah referensi para pembaca secara khusus Mahasiswa informatika dan secara umum bagi kalangan umum. Penulis menyadari bahwa dalam proses penyusunan skripsi ini melibatkan banyak pihak. Oleh sebab itu dalam kesempatan ini, penulis ingin mengucapkan banyak terima kasih kepada:

1. Kepada cinta pertama dan panutanku, Ayahanda Kamaruddin, S.Pd dan Pintu Surgaku, Ibunda Hasmawati tercinta yang senantiasa memberikan rasa sayang, didikan, materi serta do'a yang selalu di panjatkan pada Allah untuk penulis, serta memberikan pengorbanan yang tak ternilai sehingga penulis mampu menyelesaikan studi sampai sarjana.
2. Bapak Prof. Dr. H. Ambo Asse, M.Ag., sebagai Rektor Perguruan Tinggi Universitas Muhammadiyah Makassar
3. Ibu Dr.Hj.Ir. Nurnawaty,ST.,MT Selaku Dekan Fakultas Teknik Universitas Muhammadiyah Makassar
4. Bapak Muhydin A.M.Hayat S.Kom.,MT Selaku Ketua Prodi Informatika, Fakultas Teknik Universitas Muhammadiyah Makassar.



5. Ibu Titin Wahyuni S.Pd.,MT selaku Dosen Pembimbing I dan Bapak Fahrir Irhamna Rachman, S.kom., M.T selaku Dosen Pembimbing II yang senantiasa meluangkan waktu dan pikirannya untuk membimbing dan mengarahkan penulis dalam penyusunan skripsi ini.
6. Seluruh Dosen dan Staf Fakultas Teknik Universitas Muhammadiyah Makassar
7. Untuk kak Mirnawati, S.Pd terima kasih karena selalu memberikan dorongan dan motivasi hingga bisa ke tahap saat ini.
8. Untuk teman terbaik penulis 105841107319 yang selalu menemani, membantu hingga skripsi ini dapat diselesaikan dan terima kasih telah menepati janji untuk sama-sama berjuang menyelesaikan skripsinya masing-masing.
9. Kepada seluruh teman-teman angkatan 19 Teknik Informatika, khususnya 105841108819, 105841109619, 105841109919 yang telah memberikan support baik tenaga, waktu dan motivasi kepada saya untuk dapat menyelesaikan skripsi ini.
10. Isnawati, *last but now least*, ya! diri saya sendiri. Apresiasi yang sebesar-besarnya Karena bertanggung jawab dan menyelesaikan apa yang telah dimulai. Terima kasih karena terus berusaha dan tidak menyerah, serta senantiasa menikmati proses yang tidak mudah. terima kasih sudah bertahan sejauh ini.

Semoga Tuhan yang maha esa memberikan balasan yang lebih besar kepada beliau beliau, akhir kata semoga skripsi ini dapat memberikan manfaat bagi pembaca umumnya dan bagi penulis pada khususnya.

***“Billahi Fii Sabilil Haq Fastabiqul Khaerat”***

Makassar, 28 Agustus 2023

## DAFTAR ISI

HALAMAN PENGESAHAN .....	ii
HALAMAN PERSETUJUAN .....	iii
HALAMAN PERSEMBAHAN .....	iv
ABATRAK .....	v
ABSTRACT .....	vi
KATA PENGANTAR.....	viii
DAFTAR ISI .....	x
DAFTAR GAMBAR. ....	xii
DAFTAR TABEL .....	xiii
DAFTAR LAMPIRAN .....	xiv
DAFTAR ISTILAH .....	xv
BAB I PENDAHULUAN .....	1
A. Latar Belakang .....	1
B. Rumusan Masalah .....	2
C. Tujuan Penelitian .....	2
D. Manfaat Penelitian .....	2
E. Ruang Lingkup Penelitian .....	2
F. Sistematika Penelitian .....	3
BAB II TINJAUAN PUSTAKA .....	4
A. Landasan Teori .....	4
1. Ayam .....	4
2. Sistem Pakar.....	8
3. <i>Dataset Collection</i> .....	10
4. Citra Digital.....	10

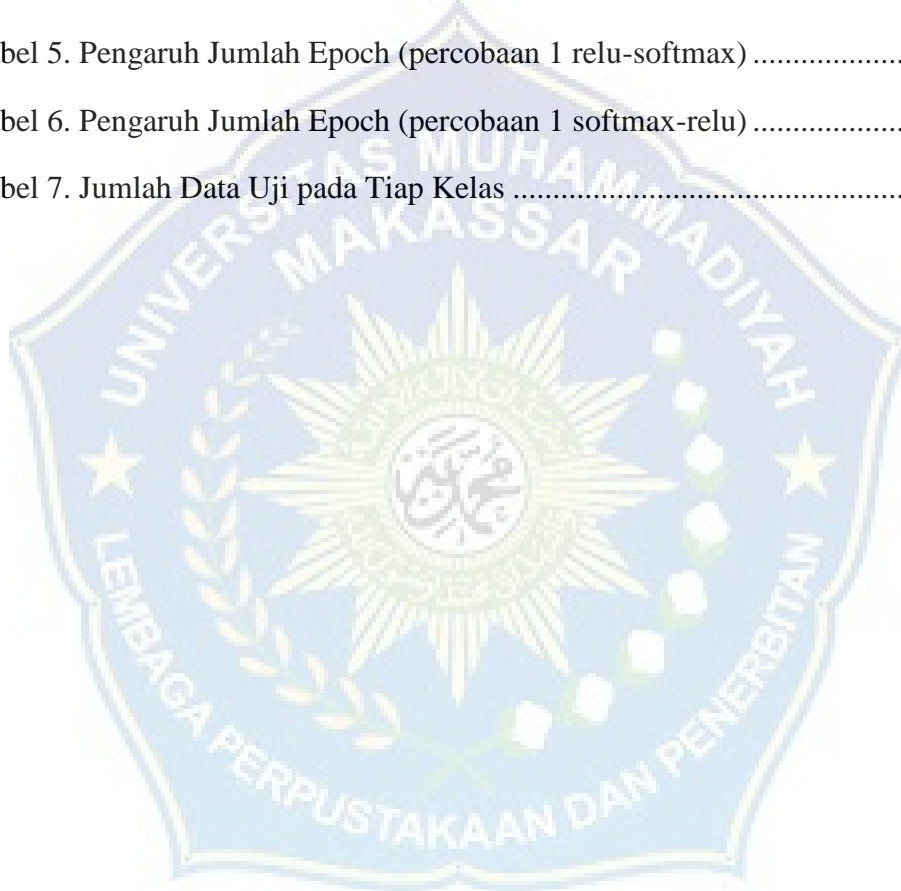
5. Pengolahan Citra Digital .....	11
6. Keras .....	11
7. <i>Tenserflow</i> .....	11
8. Phyton .....	11
9. <i>Convolutional Neural Network (CNN)</i> .....	12
10. Kaggle .....	12
11. <i>Flowchart</i> .....	12
12. Goggle Colab .....	13
13. Fungsi Aktivasi .....	13
B. Penelitian Terdahulu .....	13
C. Kerangka Pikir .....	16
BAB III METODE PENELITIAN .....	18
A. Tempat dan Waktu Penelitian .....	18
B. Alat dan Bahan .....	18
C. Perancangan Sistem .....	19
D. Teknik Analisis Data .....	21
BAB IV HASIL DAN PEMBAHASAN .....	23
A. Pengumpulan <i>Dataset</i> .....	23
B. Pembagian <i>Dataset</i> .....	24
C. Resize .....	25
D. Training Data .....	26
E. Hasil Proses <i>Training</i> .....	34
F. Load and Testing .....	36
G. Pengaruh Hyperparameters terhadap Akurasi .....	38
H. Hasil Proses Testing .....	40
BAB V KESIMPULAN DAN SARAN .....	42
A. Kesimpulan .....	42
B. Saran .....	42
DAFTAR PUSTAKA .....	44
LAMPIRAN .....	46

## DAFTAR GAMBAR

Gambar 1. Penyakit Coryza Ayam .....	6
Gambar 2. Penyakit fowl pox .....	7
Gambar 3. Penyakit Bumlefoot .....	7
Gambar 4. Arsitektur Single-channel CNN (Zhang et al., 20220) .....	12
Gambar 5. Kerangka Pikir .....	17
Gambar 6. <i>Flowchart Testing</i> .....	19
Gambar 7. <i>Flowchart Training</i> .....	20
Gambar 8. Data Gambar dari Dinas Pertanian dan Ketahanan Pangan Kota Bulukumba.....	23
Gambar 9. Data Gambar dari Lingkungan Kecamatan Bonto Tiro Kabupaten Bulukumba.....	24
Gambar 10. Struktur Direktori <i>Dataset</i> .....	24
Gambar 11. Gambar untuk Melakukan <i>resizing</i> .....	25
Gambar 12. Visualalisasi Akurasi Training dan Validasi Training <i>Accuracy</i> .....	35
Gambar 13. Sampel Data Testing.....	41

## DAFTAR TABEL

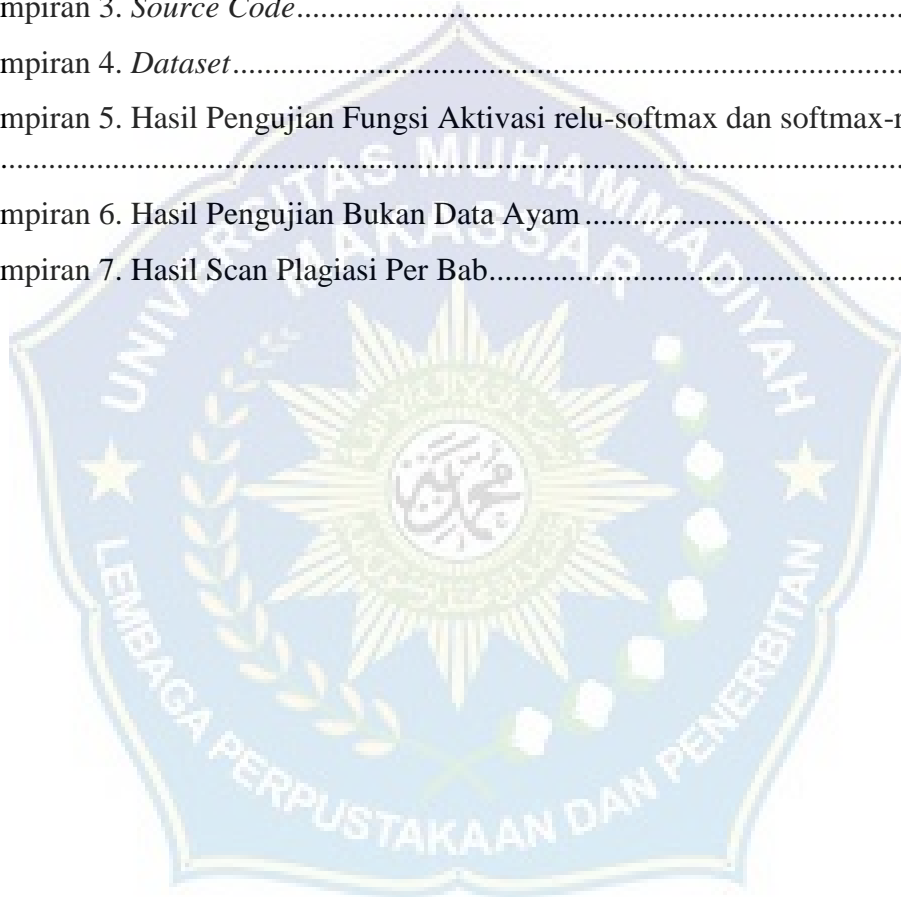
Tabel 1. Spesifikasi Komputer .....	18
Tabel 2. Sebaran Jumlah Data Citra.....	25
Tabel 3. Inisialisasi Parameter .....	29
Tabel 5. Pengaruh Jumlah Epoch (percobaan 1 relu-softmax) .....	38
Tabel 6. Pengaruh Jumlah Epoch (percobaan 1 softmax-relu) .....	39
Tabel 7. Jumlah Data Uji pada Tiap Kelas .....	40





## DAFTAR LAMPIRAN

Lampiran 1. Surat Izin Penelitian.....	47
Lampiran 2. Surat Hasil Penelitian .....	48
Lampiran 3. <i>Source Code</i> .....	49
Lampiran 4. <i>Dataset</i> .....	56
Lampiran 5. Hasil Pengujian Fungsi Aktivasi relu-softmax dan softmax-relu .....	59
Lampiran 6. Hasil Pengujian Bukan Data Ayam .....	60
Lampiran 7. Hasil Scan Plagiasi Per Bab.....	61



## DAFTAR ISTILAH

Keras	Keras adalah salah satu library pemrograman yang populer dalam bidang deep learning. Keras menyediakan antarmuka tingkat tinggi untuk membangun dan melatih model jaringan saraf tiruan (neural network).
Tenserflow	TensorFlow adalah salah satu library open-source yang populer untuk pemrograman dalam bidang machine learning dan deep learning. TensorFlow dirancang untuk mempermudah pembuatan, pelatihan, dan deploy model-model machine learning dengan menggunakan aliran data melalui grafik komputasi
Phyton	Python adalah bahasa pemrograman tingkat tinggi yang populer dan sering digunakan dalam berbagai bidang seperti pengembangan web, analisis data, kecerdasan buatan, dan lainnya.
<i>Convolution neural network</i> (CNN)	<i>Convolutional Neural Network</i> (CNN) atau dalam bahasa Indonesia dikenal sebagai Jaringan Saraf Konvolusional adalah jenis arsitektur jaringan saraf yang khusus dirancang untuk memproses data dengan struktur grid, seperti gambar dan video.
Fungsi Aktivasi	Fungsi aktivasi adalah fungsi matematika yang diterapkan pada output dari setiap neuron dalam jaringan saraf. Fungsi ini memberikan non-linearitas pada model, memungkinkannya untuk belajar pola dan hubungan yang kompleks antara input dan output.

<i>Relu</i>	ReLU ( <i>Rectified Linear Unit</i> ) adalah salah satu fungsi aktivasi yang paling umum digunakan dalam jaringan saraf.
<i>Softmax</i>	Softmax adalah fungsi aktivasi yang sering digunakan dalam jaringan saraf untuk tugas klasifikasi multikelas. Fungsi ini mengonversi output dari setiap neuron menjadi probabilitas yang berjumlah 1, sehingga memungkinkan model memberikan distribusi peluang pada setiap kelas.
<i>Train</i>	"Train" dalam konteks jaringan saraf merujuk pada proses pembelajaran atau pelatihan model menggunakan data latih (training data).
<i>Valid</i>	"Valid" dalam konteks pembelajaran mesin merujuk pada proses validasi model. Validasi digunakan untuk mengevaluasi kinerja model yang telah dilatih menggunakan data latih (training data) sebelum diuji dengan data baru.
<i>Testing</i>	"Testing" dalam konteks pembelajaran mesin merujuk pada proses pengujian atau evaluasi performa model yang telah dilatih menggunakan data latih (training data) dan divalidasi menggunakan validation set.
<i>Loss</i>	"Loss" dalam konteks pembelajaran mesin merujuk pada ukuran atau metrik yang menggambarkan seberapa besar perbedaan atau kesalahan antara prediksi model dengan nilai target yang sebenarnya.
<i>Validation accuracy</i>	"Validation accuracy" mengacu pada metrik evaluasi yang digunakan untuk mengukur sejauh mana model dapat memprediksi dengan benar pada data validasi.
<i>Citra/image</i>	Citra atau image adalah representasi visual dari objek dalam bentuk dua dimensi. Dalam konteks komputasi, citra sering kali merujuk pada data yang terdiri dari piksel-piksel berwarna

yang membentuk gambar.

Bit	Bit adalah singkatan dari "binary digit" dan merupakan unit dasar dalam komputasi digital. Bit hanya memiliki dua nilai yang mungkin, yaitu 0 atau 1. Dalam sistem bilangan biner, bit digunakan untuk merepresentasikan informasi dengan menggunakan kombinasi dari angka 0 dan 1.
<i>Epoch</i>	<i>Epoch</i> mengacu pada satu putaran lengkap saat melatih model dengan seluruh data pelatihan yang tersedia.
Byte	Byte adalah unit pengukuran dalam komputasi yang terdiri dari 8 bit. Byte digunakan untuk merepresentasikan informasi atau data dalam bentuk bilangan bulat, karakter, atau kombinasi lainnya.
Pixel	Pixel, singkatan dari "picture element", adalah unit dasar dalam representasi visual digital. Sebuah pixel merupakan titik terkecil yang membentuk sebuah gambar atau citra di layar komputer atau perangkat tampilan lainnya.
Google colab	Google Colab (singkatan dari Google Colaboratory) adalah platform cloud yang disediakan oleh Google untuk menjalankan kode Python dan melakukan pemrosesan data secara interaktif.
<i>Flowchart</i>	<i>Flowchart</i> adalah representasi visual dari alur atau urutan langkah-langkah dalam sebuah proses. Flowchart digunakan untuk menggambarkan bagaimana suatu tugas, algoritma, atau prosedur dijalankan secara sistematis.
Phyton	Python adalah bahasa pemrograman tingkat tinggi yang populer dan sering digunakan dalam pengembangan perangkat lunak, analisis data, kecerdasan buatan (AI), pembuatan aplikasi web, dan banyak lagi.





## **BAB I**

### **PENDAHULUAN**

#### **A. Latar Belakang**

Ayam merupakan ternak yang populer dikalangan masyarakat, Komoditas ayam di Indonesia mempunyai prospek pasar yang sangat baik sebab didukung oleh karakteristik produk yang bisa diterima oleh seluruh lapisan masyarakat, karena daging ayam dianggap mengandung kolesterol yang lebih rendah, dan harga daging ayam relatif lebih murah (dibandingkan dengan daging sapi juga daging kambing) dengan akses yang gampang diperoleh karena sudah menyebar pada seluruh daerah tanah air. Oleh karena itu, banyak masyarakat yang beternak ayam sebagai ladang bisnisnya. Kendala utama pada pengembangan usaha ternak ayam adalah adanya berbagai macam penyakit (Murtidjo, 1992).

Untuk memperoleh kualitas ayam yang baik serta laba yang cukup besar, peternak harus mampu memelihara dan merawat ayam supaya tidak mudah terjangkit penyakit. Kecakupan nutrisi tubuh ayam berpengaruh besar terhadap produktivitas dan hal itu sangat berkaitan erat dengan kesehatan pada ayam. Beberapa penyakit pada ayam berdampak ekonomis karena dapat mengurangi kualitas ayam yang baik sehingga merugikan peternak. Permasalahan utama yang merupakan tantangan terberat pada peternakan ayam ialah munculnya penyakit, sehingga pengelolanya perlu dilakukan secara efisien dan profesional. Namun peternak umumnya hanya mengetahui gejala-gejala yang terjadi pada ayam yang sakit, tanpa mengetahui penyakit apa yang dideritanya. Untuk dapat mendatangkan seorang pakar atau dokter hewan tidaklah mudah karena dilihat dari segi biaya cukup memberatkan peternak-peternak kecil juga membutuhkan waktu yang relatif lama untuk memanggil seorang pakar, di samping itu jumlah pakar atau dokter hewan yang sesuai dengan biaya bidangnya sangat terbatas apalagi di pedesaan.

Berdasarkan permasalahan tersebut, dapat di selesaikan menggunakan sebuah sistem pakar, dimana sistem pakar ini merupakan sistem berbasis komputer yang menggunakan pengetahuan, informasi, dan teknik penalaran

dalam memecahkan problem, yang umumnya hanya bisa diselesaikan oleh seorang ahli dibidang eksklusif. Oleh karena itu, dikembangkan suatu sistem pakar untuk mendiagnosis penyakit pada ayam dengan deteksi citra, sehingga bisa diakses oleh para peternak ayam secara online, dan bisa membantu oleh para peternak ayam agar mudah dalam mendiagnosis penyakit pada ayam. Diharapkan juga sistem pakar ini bisa menggantikan peranan seorang pakar dalam menangani penyakit pada ayam.

Berdasarkan gambaran diatas, penulis tertarik membahas mengenai sistem pakar deteksi citra pada penyakit ayam di Kecamatan Bonto Tiro Kabupaten Bulukumba.

## **B. Rumusan Masalah**

Berdasarkan uraian latar belakang di atas, maka penulis dapat merumuskan masalah sebagai berikut:

Bagaimana mengetahui hasil perbandingan fungsi aktivasi di hidden layer dan output layer?

## **C. Tujuan Penelitian**

Adapun yang menjadi tujuan dari penelitian ini adalah:

Untuk mengetahui perbandingan fungsi aktivasi di hidden layer dan output layer

## **D. Manfaat Penelitian**

Manfaat penelitian ini adalah sebagai berikut:

1. Dapat membantu mengetahui perubahan model dan akurasi mana yang lebih tinggi melalui perbandingan fungsi aktivasi.
2. Sebagai bahan referensi dalam menyikapi permasalahan pada penyakit ayam

## **E. Ruang Lingkup Penelitian**

Ruang lingkup dari penelitian ini adalah:

1. Penelitian ini membahas perbandingan fungsi aktivasi di hidden layer (relusoftmax dan softmax-relu)
2. Dataset yang digunakan sebanyak 2.271 dengan size 256x256

3. Penentuan jenis penyakit ayam dilakukan melalui deteksi citra
4. Klasifikasi citra mencakup enam kelas yaitu kaki, mata, kepala untuk gejala penyakit ayam dan kaki, mata, kepala untuk ayam sehat
5. Metode yang digunakan pada sistem pakar ini adalah CNN (Convolutional Neural Network)

## **F. Sistematika Penulisan**

### **BAB I PENDAHULUAN**

Bab ini menjelaskan mengenai latar belakang, pokok permasalahan, tujuan penulisan, batasan masalah, metoda penelitian dan sistematika penulisan

### **BAB II TINJAUAN PUSTAKA**

Bab ini membahas landasan-landasan teori sebagai hasil dari studi pustaka yang berhubungan dalam deteksi citra pada penyakit ayam, serta buku-buku, jurnal, makalah, internet, atau referensi lainnya dipergunakan sebagai acuan atas sistem pakar yang di buat.

### **BAB III METODE PENELITIAN**

Bab ini beisikan tentang permasalahan yang dihadapi, penyelesaian, masalah, metode pengembangan, *flowchart*, algoritma program, dan struktur data yang digunakan

### **BAB IV HASIL DAN PEMBAHASAN**

Bab ini berisi pembahasan mengenai temuan selama proses penelitian berlangsung serta analisis terhadap hasil perbandingan yang telah dilakukan dalam peneltian.

### **BAB V PENUTUP**

Pada bab ini berisi kesimpulan dari penulis mengenai penelitian yang telah dilakukan serta saran untuk pengembangan lebih lanjut mengenai penelitian ini supaya memperoleh hasil yang baik dikemudian hari.

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **A. Landasan Teori**

##### **1. Ayam**

Ayam (*Gallus gallus domesticus*) adalah salah satu unggas yang biasa dipelihara orang untuk dimanfaatkan sebagai keperluan hidup atau bisnis. Secara ilmu hewan digolongkan dalam keluarga phasianidae yang termasuk Gallus.

Ayam merupakan hewan peliharaan yang adaptif dan dapat dikatakan bisa hidup di sembarangan tempat, asalakan tersedianya makanan. Namun tak sedikit juga ayam yang terserang penyakit yang dapat merugikan para peternak ayam sehingga mengakibatkan kerugian.

##### **a. Penyebab Penyakit Ayam**

Ayam bisa terserang penyakit karena faktor lingkungan atau kondisi tubuh ayam yang bersangkutan. Secara umum, penyebab tersebut dikategorikan ke dalam tiga faktor, yaitu infeksi, perkembangan tingkat resistensi, dan penyakit yang tersembunyi.

##### **(1) Infeksi**

Infeksi biasanya disebabkan oleh mikroorganisme, seperti virus atau bakteri. Mikroorganisme yang menyebabkan ayam sakit merupakan bibit penyakit yang bersifat pathogen (*pathogenic microorganism*). Rendah atau kurangnya sistem kekebalan tubuh menyebabkan mikroorganisme pathogen tersebut dengan mudah menyerang beberapa bagian tubuh sehingga ayam tersebut terkena penyakit. Tingkat keganasan penyakit, penyebaran, dan tingkat kematian yang diakibatkan biasanya berbeda-beda tergantung pada tingginya tingkat infeksi yang terjadi.

##### **(2) Perkembangan Tingkat Resistensi**

Mikroorganisme salah satu penyebab penyakit berada di dalam tubuh ayam, dengan sendirinya ayam akan membuat rangkaian senyawa kimiawi

yang dapat bertahan terhadap anti bodi yang dihasilkan dalam tubuh. Semakin lama organisme tersebut akan semakin bisa bertahan dan berkembang dalam tubuh. Akibatnya, ayam bisa dengan mudah terserang oleh penyakit tersebut.

### (3) Penyakit yang tersembunyi (*Disease Subsides*)

Secara alami, zat anti bodi dalam tubuh ayam akan melakukan perlawanan dan penghancuran terhadap bibit penyakit yang masuk ke dalam aliran darah. Namun pada kenyataannya, banyak kasus bibit penyakit itu tidak tuntas dimusnahkan. Akibatnya bibit penyakit masih ada dalam tubuh dengan tingkat yang rendah dan tidak terdeteksi. Ayam yang terinfeksi tampak sehat, padahal bibit penyakit yang tersembunyi bisa kapan saja berkembang dan menyerang tubuh ayam ketika kondisi pertahanan tubuhnya menurun.

### **b. Ciri-Ciri Ayam Sakit**

Mengetahui gejala atau tanda-tanda umum ayam sakit merupakan salah satu metode pemeliharaan ayam. Dengan mengetahui apa saja tanda-tanda ayam sakit maka peternak dapat mengatasi masalah penyakit ayam. Sehingga penanganan masalah tentang penyakit ayam nantinya berjalan dengan tepat, efektif dan efisien. Sebagai peternak yang perhatian dengan ternaknya maka mengetahui gejala yang terjadi dalam ayam adalah hal utama (Titik Sudaryani, 2005).

Ciri-ciri fisik ayam sakit menurut Titik Sudaryani (2005):

#### (1) Kepala

Kepala adalah bagian tubuh ayam yang paling sering diamati. Pada ayam yang sakit umumnya memasukkan kepala di bawah sayap. Jengger tidak merah merona, pucat, hidung berair, terkadang pilek dan mata kering, sedikit sayup, serta mulut menganga atau seing ngorok.

#### (2) Bulu

Bulu mengembang, dan terlihat bundar

#### (3) Kaki

Pada kaki ayam sedikit dilipat dan tidak tegak. Akibatnya ayam sedikit lebih pendek dari ayam sehat lainnya.

#### (4) Perut



Pada bagian perut sangat kencang bahkan halus. Ini diakibatkan ayam banyak mengonsumsi air.

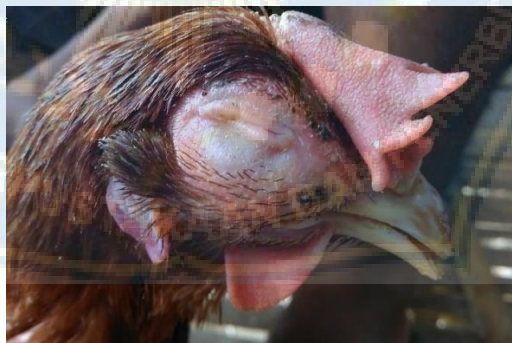
(5) Kotoran

Ayam yang sakit kotorannya lebih encer biasanya berwarna hijau, merah, kuning, dan putih.

**c. Penyakit Ayam**

(1) *Infectious coryza* (snot)

*Infectious coryza* (snot) merupakan penyakit pernafasan bagian atas pada unggas, terutama ayam. Penyakit ini telah menyebar luas di dunia, dan seringkali terjadi pada musim dingin atau udara jelek. *Infectious coryza* (snot) merupakan penyakit pernafasan bagian atas pada unggas, terutama ayam, yang bersifat akut. Penyakit ini disebabkan oleh bakteri berbentuk batang yang pleomorfik tidak bergerak, bersifat gram negatif dan disebut *Hemophilus gallinarum*. Sifat alami bakteri ini tidak bisa hidup lama, dan akan mati dalam waktu 4-5 jam. Penularan penyakit dapat terjadi melalui kontak langsung dengan ayam sakit atau ayam karier, dapat pula terjadi secara tidak langsung melalui air minum, pakan, dan peralatan yang terkontaminasi.

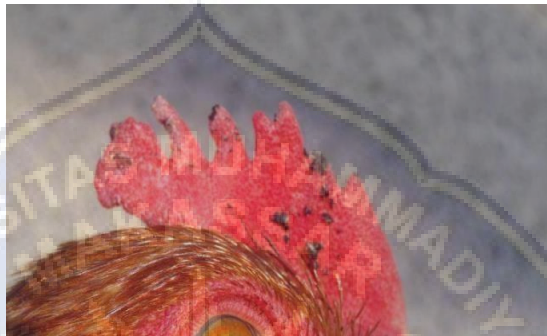


**Gambar 1.** Penyakit Coryza

Ayam ras atau ayam pedaging, terutama pada jenis-jenis tertentu seperti ayam broiler dan ayam kampung merupakan jenis ayam yang sering terserang penyakit coryza dan pencegahan penyebaran penyakit ini yaitu pentingnya menjaga kebersihan kandang dan lingkungan sekitar. Vaksinasi rutin juga merupakan salah satu langkah pencegahan penyakit ini terjadi.

## (2) Fowl pox (cacar unggas)

Penyebab terjadinya cacar unggas disebabkan oleh virus Poxvirus, biasanya ayam yang terserang penyakit ini ditandai dengan papula kecil berwarna kelabu dan kulit tidak berbulu pada bagian kepala atau jengger. Penyakit ini dapat ditularkan secara langsung melalui kontak dengan luka di kulit atau bisa juga secara tidak langsung yaitu melalui udara, serta penyebaran pada makanan yang tersrang penyakit.



**Gambar 2.** Penyakit fowl pox

Jenis ayam yang paling umum erserang penyakit fowl pox adalah berbagai jenis ayam domestik, termasuk ayam petelur dan ayam pedaging. Upaya yang dilakukan dalam pencegahan penyakit ini dengan menjaga kebersihan kandang dan lingkungan sekitar dan melakukan vaksinasi rutin yang merupakan langkah preventif dalam melindungi ternak dari infeksi fowl pox.

## (3) Bumble foot (infeksi telapak kaki ayam)

*Bumble Foot* umumnya menyerang ayam jantan, walaupun dapat juga menyerang ayam betina, baik di usia muda hingga dewasa. Penyakit ini menyebabkan penebalan pada telapak kaki dan kaki bengkak. Penyakit ini disebabkan oleh bakteri *Staphylococcus*. Infeksi terjadi karena adanya luka pada telapak kaki ayam, baik karena goresan maupun tertusuk benda lainnya.



Gambar 3. Penyakit Bumlefoot

Jenis ayam yang sering terserang penyakit Bumlefoot adalah ayam peliharaan atau ayam petelur. Penyakit ini terjadi pada ayam yang dipelihara dalam kondisi kurang bersih, dan perawatan kaki dan lingkungan yang kurang memadai. Untuk mencegah penyakit ini terjadi yaitu menjaga kebersihan kandang dan area tempat ayam berkeliaran.

## 2. Sistem Pakar

### a. Pengertian Sistem Pakar

Sistem pakar atau *Expert System* biasa disebut juga dengan *Knowledge Based System* adalah aplikasi komputer yang dapat membantu dalam pengambilan keputusan atau pemecahan masalah dalam bidang yang spesifik. Sistem pakar ini dapat bekerja dengan metode analisis yang terlebih dahulu didefinisikan oleh pakar yang sesuai dengan bidang keahliannya.

Adapun pengertian sistem pakar menurut beberapa ahli yakni:

Sistem pakar adalah suatu cabang dari AI yang membuat penggunaan secara luas pengetahuan yang khusus untuk penyelesaian masalah tingkat manusia yang pakar menurut Arhami, 2004

Sistem pakar utau cabang dari kecerdasan buatan yang cukup tua karena sistem ini mulai dikembangkan pada tahun 1960. Sistem pakar adalah program kecerdasan buatan dengan basis pengetahuan yang diperoleh dari pengalaman atau pengetahuan pakar atau ahli dalam memecahkan persoalan dalam bidang tertentu dan didukung mesin *Interensi* yang melakukan penalaran atau pelacakan terhadap sesuatu atau fakta-fakta atau aturan kaidah yang ada di basis pengetahuan setelah dilakukan pencarian sehingga dicapai kesimpulan menurut siswanto, 2010.

Sistem pakar merupakan salah satu bidang teknik kecerdasan buatan yang cukup diminati karena penerapannya diberbagai bidang baik ilmu pengetahuan maupun bisnis yang terbukti sangat membantu dalam mengambil keputusan dan sangat luas penerapannya. Sistem pakar adalah suatu sistem computer yang dirancang agar dapat dilakukan penalarannya seperti layaknya seorang pakar pada suatu bidang keahlian tertentu.

## **b. Manfaat dan Kekurangan Sistem Pakar**

### **(1) Manfaat Sistem Pakar**

Sistem Pakar (Expert System) adalah sistem yang dirancang untuk dapat menirukan keahlian seorang pakar dalam menjawab pernyataan dan memecahkan suatu masalah (T. Sutojo, dkk;2011:13).

- (a) Meningkatkan produktivitas, karena sistem pakar dapat bekerja lebih cepat daripada manusia.
- (b) Membuat seorang yang awam bekerja seperti layaknya seorang pakar.
- (c) Meningkatkan kualitas, dengan memberi nasehat yang konsisten dan mengurangi kesalahan.
- (d) Mampu menangkap pengetahuan dan kepakaran seseorang.
- (e) Memudahkan akses pengetahuan seorang pakar.
- (f) Bisa digunakan sebagai media perlengkapan dalam pelatihan. Pengguna pemula yang bekerja dengan sistem pakar akan menjadi lebih berpengalaman karena adanya fasilitas penjelas yang berfungsi sebagai guru.
- (g) Meningkatkan kemampuan untuk menyelesaikan masalah karena sistem pakar mengambil sumber pengetahuan dari banyak pakar.

### **(2) Kekurangan Sistem Pakar**

Selain manfaat ada juga beberapa kekurangan yang ada pada Sistem Pakar, diantaranya:

- (a) Biaya yang sangat mahal untuk membuat dan memeliharanya.
- (b) Sulit dikembangkan karena keterbatasan keahlian dan ketersediaan pakar

(c) Sistem pakar tidak 100% bernilai benar.

### **c. Komponen Sistem Pakar**

Sebuah program sistem pakar terdiri atas beberapa komponen yang mutlak harus ada. Komponen itu adalah sebagai berikut:

#### **(1) Basis Pengetahuan (*Knowledge Base*)**

Basis pengetahuan merupakan inti program sistem pakar karena basis pengetahuan ini merupakan representasi pengetahuan (*Knowledge Representation*) dari seorang pakar.

#### **(2) Basis Data**

Basis data adalah bagian yang menandung semua fakta, baik pada fakta awal pada saat sistem mulai beroperasi maupun fakta yang didapatkan pada saat pengambilan kesimpulan sedang dilaksanakan

#### **(3) *Mesin Inferensi***

*Mesin Inferensi* adalah bagian yang mengandung mekanisme fungsi berikir dan pola penalaran sistem yang digunakan oleh seorang pakar. Mekanisme ini akan menganalisa suatu masalah tertentu dan selanjutnya akan mencari jawaban atau kesimpulan yang terbaik. *Mesin Inferensi* memulai pelacakannya dengan mencocokkan kaidah dalam basis pengetahuan dengan fakta yang ada dalam basis data.

#### **(4) Antar Muka Pemakai (*User Interface*)**

Antar muka pemakai adalah bagian penghubung antara program sistem pakar dengan pemakainya. Pada bagian ini akan terjadi dialog antara program dengan pemakai.

### **3. Dataset Collection**

*Dataset Collection* merupakan tahapan pengumpulan data yang akan digunakan pada penelitian ini dalam menentukan penyakit pada ayam. Dataset yang digunakan pada penelitian ini berupa gambar penyakit dari luar yang menyerang ayam contohnya penyakit snot atau coryza, penyakit cacar, dan bubul ayam.



#### 4. Citra Digital

Citra digital adalah gambar dua dimensi yang dihasilkan dari gambar analog dua dimensi yang kontinu menjadi gambar diskrit melalui proses *sampling*. Dalam citra digital terdapat sebuah objek larik (array) yang berisi nilai-nilai *real* dan kompleks yang direpresentasikan dengan deretan bit tertentu.

#### 5. Pengolahan Citra Digital

Pengolahan citra digital (*Digital Image Processing*) adalah ilmu yang mempelajari tentang teknik-teknik dalam mengolah citra. Citra yang dimaksud adalah gambar diam (foto) maupun gambar bergerak (video). Sedangkan digital adalah pengolahan citra atau gambar dilakukan secara digital menggunakan computer. Tujuan dari pengolahan citra ini adalah memperbaiki kualitas suatu citra sehingga dapat diinterpretasi dengan mudah oleh manusia atau sebuah mesin (computer).

#### 6. Keras

Keras merupakan library deep learning jaringan saraf tiruan tingkat tinggi yang ditulis dengan Bahasa python dan mampu berjalan diatas TensorFlow, CNTK dan Theano. Library ini menyediakan fitur yang digunakan dengan fokus mempermudah pengembangan lebih dalam tentang deep learning. Library ini dikembangkan agar memungkinkan pengujian yang cepat pada CPU dan GPU serta mendukung algoritma convolutional neural network dan recurrent neural network ataupun kombinasi dari keduanya.

#### 7. Tensorflow

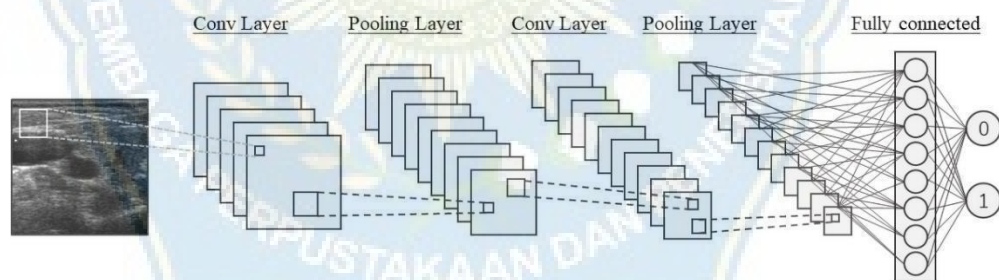
Tensorflow merupakan salah satu *framework deep learning* dan juga salah satu *library* untuk data *science* yang bersifat *free open source* yang telah dikembangkan oleh para peneliti dari tim google. Tensorflow juga dapat digunakan dalam berbagai bidang. Dalam bidang *object detection* terdapat *framework tensorflow object detection API* yang merupakan alat yang dapat digunakan untuk mempermudah proses *constructing*, *training* dan *deployment* pada suatu model *object detection*. *Framework tensorflow object detection API* menyediakan *pretrained object detection model* bagi *user*.

## 8. Python

Python adalah bahasa pemrograman interpretative multiguna dengan filosofi perancangan yang berfokus pada tingkat keterbacaan kode. Python merupakan bahasa yang menggabungkan kapasitas, kemampuan, dengan sintaksi kode yang sangat jelas, dan dilengkapi dengan fungsionalitas pustaka standar yang besar dan komprehensif. Python adalah bahasa pemrograman dengan tujuan umum yang kemudian dikembangkan secara khusus untuk membuat *source code* mudah dibaca. Python juga memiliki *library* yang lengkap sehingga memungkinkan *programmer* untuk membuat aplikasi yang mutakhir dengan menggunakan *source code* yang tampak sederhana (Ljubomir Perkovic, 2012).

## 9. Convolutional Neural Network (CNN)

Metode Convolutional Neural Network merupakan salah satu metode dari deep learning yang dapat melakukan proses pembelajaran mandiri untuk mengenali suatu objek, ekstraksi objek dan klasifikasi serta dapat diterapkan pada citra resolusi tinggi yang memiliki model distribusi non-parametrik (Zhang et al., 2018 dalam Arrofiqoh dan Harintaka, 2018).



Gambar 4. Arsitektur Single-channel CNN (Zhang et al., 20220)

## 10. Kaggle

Kaggle merupakan *platform online* yang mempunyai kumpulan dataset yang digunakan baik individu maupun kelompok untuk memodelkan data dan memperoleh suatu data.

## 11. Flowchart

Bagian alir (*flowchart*) adalah bagan (*chart*) yang menunjukkan aliran (*flow*) di dalam program atau prosedur sistem secara logika, digunakan sebagai

alat bantu komunikasi dan dokumentasi. Bagan alur sistem (*System flowchart*) merupakan bagan yang menunjukkan arus pekerjaan dari sistem secara keeluruhan, menjelaskan urutan dari prosedur yang ada di dalam sistem serta menunjukkan apa yang dikerjakan sistem.

## 12. Google Colab

Google Colab environment dari Google yang digunakan untuk menjalankan program. Colab adalah salah satu produk dari Google yang berbasis cloud dan dijalankan melalui browser. Colab menyediakan processor dengan spesifikasi tinggi (GPU dan TPU) dengan tujuan memudahkan para researcher menjalankan program yang membutuhkan spesifikasi tinggi secara online (Bisong, 2019). Google menyediakan GPU (graphics processing unit) yang bisa digunakan secara gratis dengan tipe Nvidia K80s, P4s, T4s, dan untuk versi berbayar akan mendapatkan tipe V100 dan P100s.

## 13. Fungsi Aktivasi

Fungsi aktivasi harus efisien secara komputasi dikarenakan harus menghitung banyak neuron untuk setiap sampel data. Fungsi aktivasi juga dapat membantu proses normalisasi output pada setiap neuron untuk memiliki nilai dalam range antara 1 dan 0, atau antara -1 dan 1. Terdapat berbagai jenis fungsi aktivasi, sebagai berikut. Fungsi aktivasi yang digunakan pada penelitian ini:

- a ReLU memiliki kepanjangan, yaitu *Rectified Linear Unit* dimana memiliki keuntungan terhadap fungsi aktivasi sigmoid dan tanh karena tidak pernah menjadi saturated (jenuh) terhadap nilai  $x$  yang besar. Dimana rata-rata dari fungsi ini tidak bernilai nol.
- b Softmax memiliki keuntungan utama adalah rentang probabilitas output dengan nilai 0 hingga 1, dan jumlah semua probabilitasnya akan sama dengan satu. Fungsi softmax digunakan untuk sebuah model multiklasifikasi yang akan mengembalikan sebuah peluang dari masing-masing kelas dan kelas target akan memiliki probabilitas yang tinggi.

## **B. Penelitian Terkait**

Penelitian terdahulu adalah upaya peneliti untuk mencari perbandingan dan selanjutnya untuk menemukan inspirasi baru untuk penelitian selanjutnya di samping itu kajian terdahulu membantu penelitian dalam memosisikan penelitian serta menunjukkan orsionalitas dari penelitian. Pada bagian ini peneliti mencamtumkan berbagai hasil penelitian terdahul yang terkait dengan penelitian yang hendak dilakukan, kemudian membuat ringkasannya, baik penelitian yang sudah terpublikasi atau belum terpublikasi (skripsi, tesis, disertasi dan sebagainya).

### **1. Helviani Zebua (2023). "Pembuatan Model Deteksi Gejala Awal Penyakit Mulut dan Kuku Pada Sapi Berbasis Citra Mengguakan Metode *Convolutional Neural Network*"**

Penyakit Mulut dan Kuku (PMK) merupakan penyakit yang sangat menular dan disebabkan oleh virus penyakit mulut dan kuku, yaitu Foot and Mouth Disease Virus (FMDV) pada hewan berkuku belah. Penyebaran PMK sangat rentang terjadi yang disebabkan oleh benda yang terkontaminasi virus PMK. Untuk meminimalisir penyebaran PMK, maka diperlukan pendeteksian PMK. Dalam kasus pendeteksian objek metodeal yang dapat digunakan adalah *Convulation Neural Network* (CNN). CNN bisa bekerja dengan baik dalam pembuatan model dengan dataset citra dibandingkan dengan algortima neural network lainnya karena jaringan yang ada pada metode CNN mengkhususkan neuron untuk mengenali bentuk citra. Penelitian ini dilakukan untuk menemukan arsitektur model deteksi gejala aal PMK dengan menggunkana metode CNN. Hasil penelitian menunjukkan bahwa ephco dan nilai batch size dapat mempengaruhi akurasi dari CNN, jumlah ephco 20 dan nilai batch size 32 menghasilkan akurasi terbaik dengan nilai akurasi training sebesar 99% dan akurasi pengujian pada 22 citra adalah sebesar 91%.

### **2. Aion Suharis Widodo (2022) "Sistem Pakar Penyakit Daun Kacang Tanah menggunakan CNN (*Convolutional Neural Network*)" Program Studi Teknik Informatika.**

Proses identifikasi deteksi citra penyakit daun tanaman kacang tanah menggunakan *Convolution Neural Network* (CNN) adalah salah satu metode dalam pengolahan citra digital untuk mendeteksi dan mengenali objek pada sebuah gambar. Salah satu algoritma yang digunakan adalah metode CNN. Algoritma CNN berfungsi mengelompokkan citra data masukan dalam beberapa kelompok berdasarkan jarak minimum. Implementasi algoritma CNN dalam mengklasifikasikan citra daun kacang yang terserang penyakit dilakukan dengan mencari rancangan arsitektur terbaik dengan membandingkan beberapa parameter dalam mengklasifikasi citra jenis penyakit Angular Leaf Spot, Bean Rust dan Kacang sehat adalah dengan menggunakan parameter size 40x40 piksel, ukuran kernel 3x3, learning rate 0,001, jenis optimizer Adam, epoch 120, batch size 30 dan scenario perbandingan dataset 90% :10% dengan jenis citra RGB (berwarna). Hasil tingkat akurasi yang diperoleh dari data testing dengan menggunakan model arsitektur terbaik dalam pengklasifikasian citra jenis penyakit pada kacang berdasarkan daunnya yaitu sebesar 80.00%.

**3. Robiatul Adawiyah. (2021). "SISTEM PAKAR DIAGNOSA PENYAKIT PADA AYAM PETELUR DENGAN METODE NAÏVE BAYES". Program Studi Teknik Informatika.**

Para peternak ayam sering sekali dihadapkan berbagai macam penyakit ayam. Penyakit pada ayam memiliki banyak jenis tetapi mempunyai beberapa gejala yang mirip dengan perbedaan yang sedikit. Untuk mempermudah peternak ayam dalam mendiagnosa penyakit pada ayam petelur, maka dibuatkan sistem pakar untuk mendeteksi penyakit pada ayam petelur. Tujuan dari pembuatan sistem pakar ini adalah membantu para peternak ayam maupun masyarakat secara umum dalam mengidentifikasi penyakit pada ayam petelur sehingga meminimalisir kerugian peternak akibat penyakit ayam yang mati karena terjangkit penyakit. Metode yang digunakan pada sistem pakar ini yaitu menggunakan metode naïve bayes. Variabel yang dibutuhkan dalam penelitian ini adalah gejala-gejala klinis pada ayam. Hasil pengujian penelitian ini menunjukkan akurasi sistem sebesar 86,7%



**4. Yegi Fahrianwari Putra. (2020). "Aplikasi Sistem Pakar Diagnosa Penyakit Pada Pernafasan Ayam Broiler Menggunakan Metode Forward Chaining Berbasis Website". Pogram Studi Teknik Informaika.**

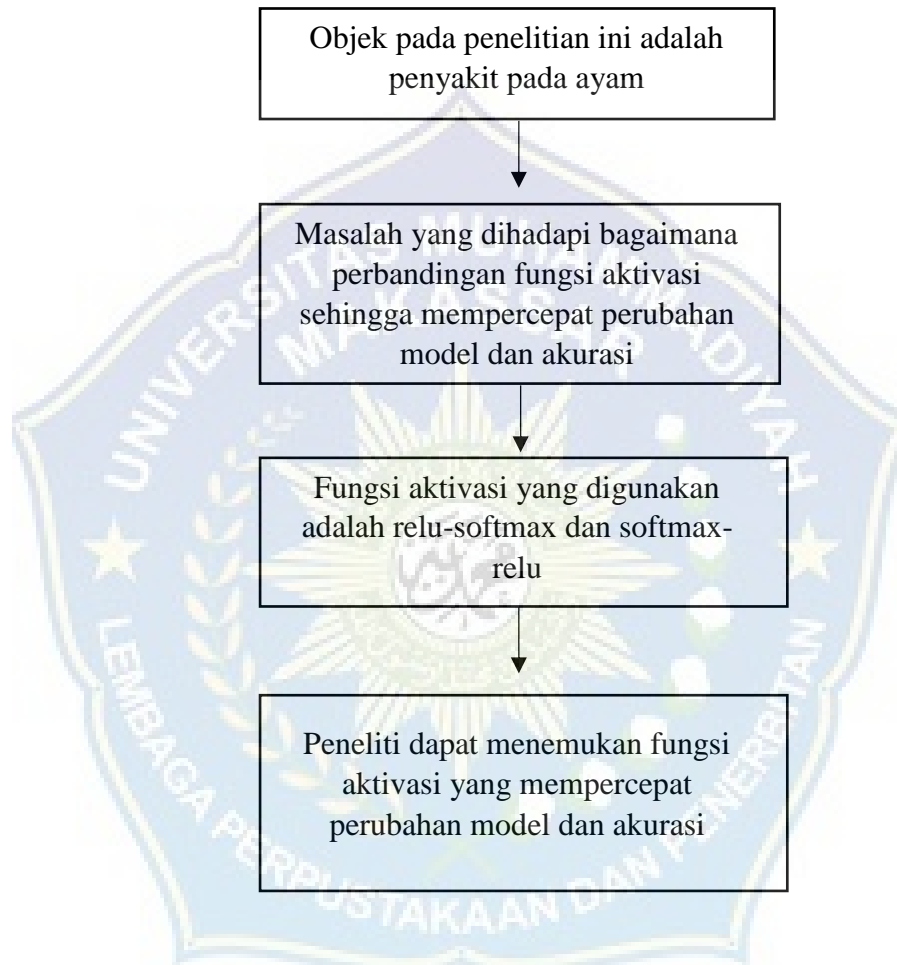
Sistem pakar merupakan aplikasi berbasis komputer yang digunakan untuk menyelesaikan masalah, penerapan sistem pakar mengidentifikasi penyakit ayam. Menurutnya produksi telur ayam dipengaruhi oleh beberapa faktor diantaranya umur, musim, penyakit, dan sistem pemeliharaan. Kurangnya pengetahuan tentang penyakit dan penyebaran dokter hewan yang tidak merata, maka diperlukan suatu sistem yang dapat memberikan rekomendasi tentang penyakit ayam buras.

**5. Ida Royani Simanungkalit (2020) "Sistem Pakar mendiagnosa Penyakit Ayam Kampung dengan Menggunakan Metode Certainty Factor berbasis web" Jurnal Nasional Komputasi dan Teknologi Informasi.**

Budidaya ternak unggas di Indonesia masih belum berhasil dalam memenuhi sebagian dari kebutuhan dalam negeri, termasuk rentan terhadap serangan penyakit hewan berbahaya. Masyarakat juga resah karena penyakit pada ayam bisa saja menular pada manusia akibat mengonsumsi daging ayam yang sudah terjangkit penyakit. Studi kasus dalam melakukan penelitian untuk mendiagnosa penyakit ayam kampung ini dilakukan pada ternak ayam kampung Bang Ari di jalan jamin gintig komplek pamen No. 14 Padang Bulan Kecamatan Medan Baru Kota Medan. Dari kriteria yang dibutuhkan merupakan jenis penyakit serta gejala-gejala penyakit yang menyeang ayam kampung serta tindakan pengobatan yang akan dilakukan terhadap ayam kampung yang terjangkit penyakit. Penelitian ini bertujuan membangun sistem pakar berbasis web dalam mendiagnosa penyakit ayam kampung menggunakan metode certainty factor. Dari contoh kasus, hasil perhitungan certainty factor pada gejala ayam kampung, maka penyakit yang memiliki nilai persentase terbesar adalah P 7 yaitu penyakit batuk manuhun sebesar 97.50%. sistem untuk mendiagnosa penyakit ayam kampung dibangun menggunakan Bahasa pemrograman berbasis web dan database MySQL.

### C. Kerangka Pikir

Untuk menyelesaikan masalah pada bab sebelumnya maka penulis membuat gambaran singkat sebagai alur penyusunan pada laporan ini dengan kerangka pemikir sebagai berikut:



**Gambar 5.** Kerangka Pikir



## **BAB III**

### **METODE PENELITIAN**

#### **A. Tempat dan Waktu Penelitian**

Pada penelitian ini penulis akan melakukan penelitian di Kecamatan Bonto Tiro, Kabupaten Bulukumba, Sulawesi Selatan. Waktu penelitian ini  $\leq$  3 bulan, terhitung pada bulan Mei-Juli 2023

#### **B. Alat dan Bahan Penelitian**

Penelitian ini membutuhkan data gambar ayam yang terjangkit penyakit dan yang tidak terjangkit penyakit. Data gambar diambil melalui dinas pertanian dan ketahanan pangan Kota Bulukumba dan diambil secara langsung dengan menggunakan *device* seperti telepon seluler.

Pada laporan ini dibutuhkan dua data yaitu data primer dan sekunder. Dimana data primer ini diambil secara langsung dan data sekunder diambil melalui dinas peternakan. Berikut adalah beberapa gambar yang di ambil secara langsung menggunakan smartphone.

Alat yang dibutuhkan dalam penelitian ini adalah sebagai berikut:

##### **1. Perangkat Lunak**

- *Michrosof Windows 10 64 bit*
- *Michroso Office word 10 64 bit*
- *Michrosof Office Excel 10 64 bit*
- *Visual Studio Code*
- Keras
- Transflow
- Phyton
- *Kaggle*
- Google Colab

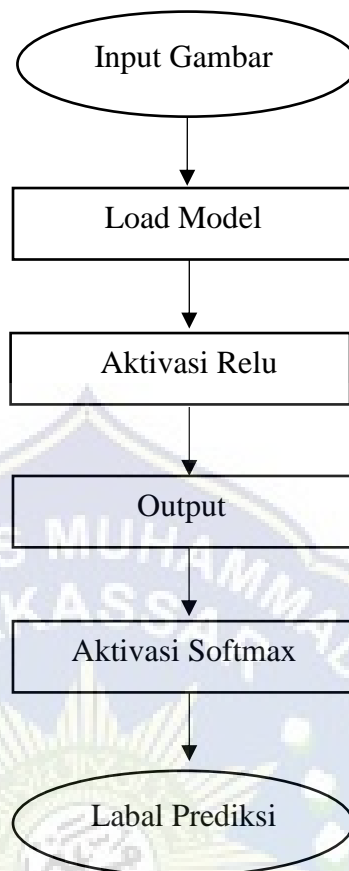
## 2. Perangkat Keras (*Hardware*)

**Table 1.** Spesifikasi Komputer

Spesifikasi	Keterangan
<i>Processor</i>	Intel(R) Quad Core i7-8550U Generasi Ke-8
RAM	4 GB
<i>Storage</i>	1 TB
<i>Graphic Card</i>	Intel® UHD Graphic 600
<i>Operating System</i>	Windows 10

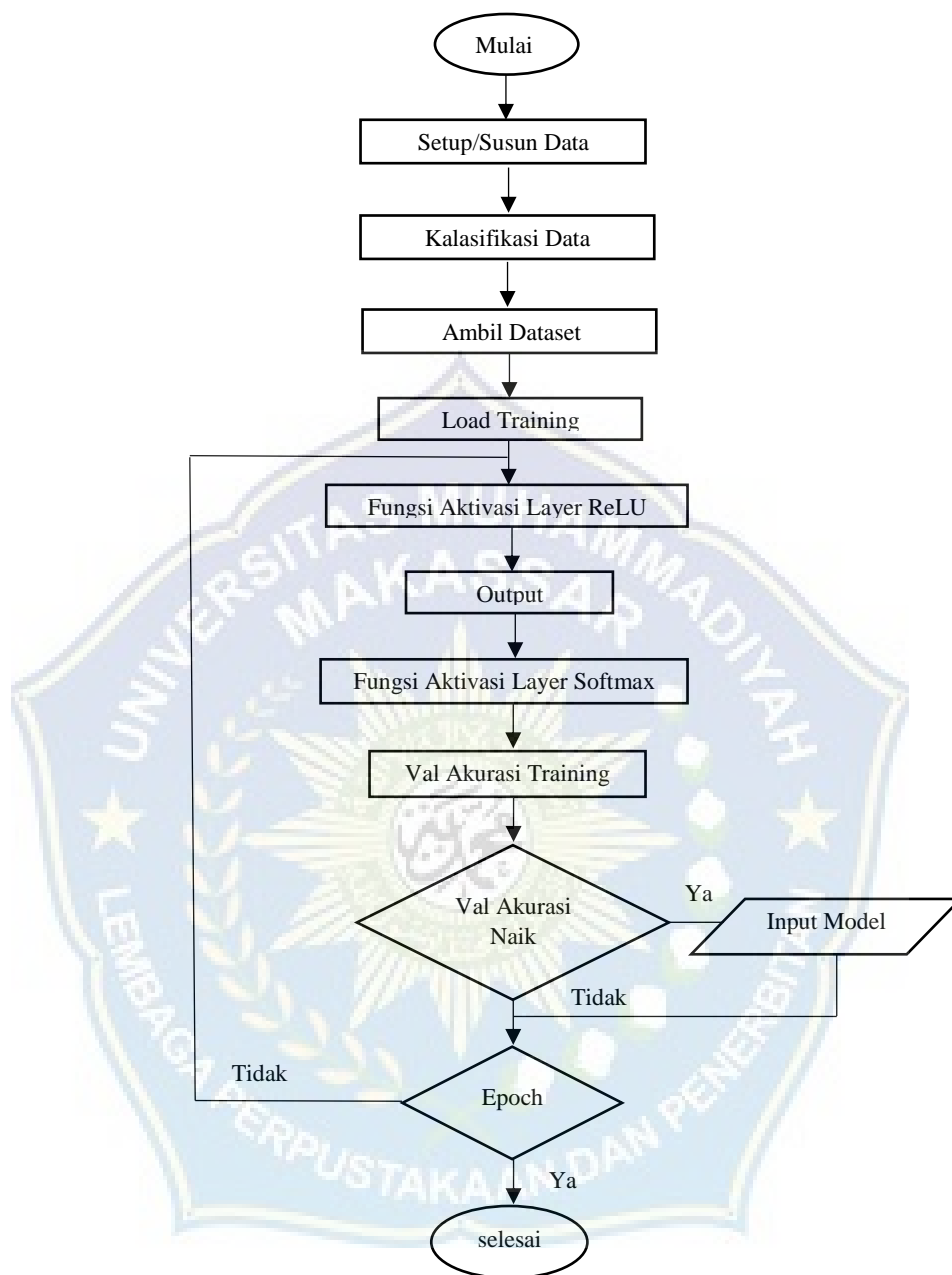
### C. Perancangan Sistem

Berikut merupakan diagram alir dari tahapan-tahapan penelitian yang akan dilakukan untuk melakukan perbandingan fungsi aktivasi



**Gambar 6.** *Flowchart Testing*

Pada gambar 6. *Flowchart Testing* perancangan di mulai dari menginput data gambar kemudain load model, setelah itu masuklah pada fungsi aktivasi, fungsi aktivasi yang di gunakan yaitu relu lalu output dan softmax, lalu akan muncul label prediksi.



**Gambar 7.** *Flowchart Training*

Perancangan sistem ini dimulai lalu menyusun data yang diambil diseluruh dataset lalu lanjut pada model klasifikasi, setelah diklasifikasi di load kembali datasetnya untuk training, lalu fungsi aktivasi relu untuk mengatasi hilangnya gradient yang terjadi pada fungsi softmax, lalu output atau keuaran dari sistem, dan fungsi aktivasi softmax digunakan pada lapisan output, setelah itu akan

muncul validasi akurasi training, jika validasi akurasi rendah maka dapat menginput model kembali, namun jika akurasi validasi rendah maka bisa menambah jumlah epoch.

#### **D. Teknik Analisis Data**

Analisis data merupakan proses mencari dan menyusun secara sistematis data yang diperoleh dari hasil wawancara, observasi, dan dokumentasi, dengan cara mengorganisasikan data ke dalam kategori, menjabarkan ke dalam unit-unit, melakukan sintesa, menyusun ke dalam pola, memilih mana yang penting dan akan dipelajari serta membuat kesimpulan agar mudah dipahami oleh diri sendiri dan orang lain.

Untuk mencapai hasil yang diinginkan maka peneliti melakukan beberapa tahapan analisis sebagai berikut:

##### **1. Reduksi Data**

Menurut Sugiyono (2016: 247), reduksi data adalah merangkum, memilih hal-hal yang pokok, memfokuskan pada hal-hal yang penting, dicari tema dan polanya. Dengan kata lain, peneliti merangkum kembali data-data untuk memilih dan memfokuskan pada bagian yang penting dan memberikan gambaran yang jelas mengenai hubungan jarak jauh yang menggunakan video call sebagai media komunikasi.

##### **2. Penyajian Data**

Penyajian data bisa dilakukan dalam bentuk uraian singkat, bagan, hubungan antar kategori, flowchart, dan sejenisnya. Yang paling sering digunakan untuk menyajikan data dalam penelitian kualitatif adalah dengan teks yang bersifat naratif (Sugiyono, 2016:249).

##### **3. Kesimpulan**

Kesimpulan adalah langkah terakhir dari suatu periode penelitian yang berupa jawaban terhadap rumusan masalah (Sugiyono, 2016:17). Pada bagian ini peneliti mengutarakan kesimpulan atas data-data yang telah

diperoleh dari hasil wawancara dan observasi, sehingga menjadi penelitian yang data menjawab permasalahan yang ada.



#### **BAB IV**

#### **HASIL DAN PEMBAHASAN**

## A. Pengumpulan *Dataset*

Pada tahap awal penelitian, penulis mengumpulkan *dataset* gejala luka pada ayam yang terjangkit penyakit ayam yaitu coryza, bumlefoot, dan fowl pox dan sehat pada bagian kaki, mata, dan jengger ayam. Jumlah keseluruhan data yang berhasil dikumpulkan penulis adalah sebanyak 2.271 data gambar dan data gambar ini diperoleh dari beberapa sumber, yaitu diperoleh di Dinas Pertanian dan Ketahanan Pangan Kota Bulukumba, dan Lingkungan Kecamatan Bonto Tiro Kabupaten Bulukumba.

### 1. Dinas Pertanian dan Ketahanan Pangan Kota Bukukumba

Data yang didapatkan dari Dinas Pertanian dan Ketahanan Pangan Kota Bulukumba merupakan data gambar gejala luka pada ayam yang terjangkit penyakit coryza, flow fox, dan bumle foot. Data gambar diambil oleh dokter hewan di dinas setempat, pada awal hingga pertengahan tahun 2021. Data yang didapatkan dari Dinas Pertanian dan Ketahanan Pangan Kota Bulukumba adalah sebanyak 998 gambar.



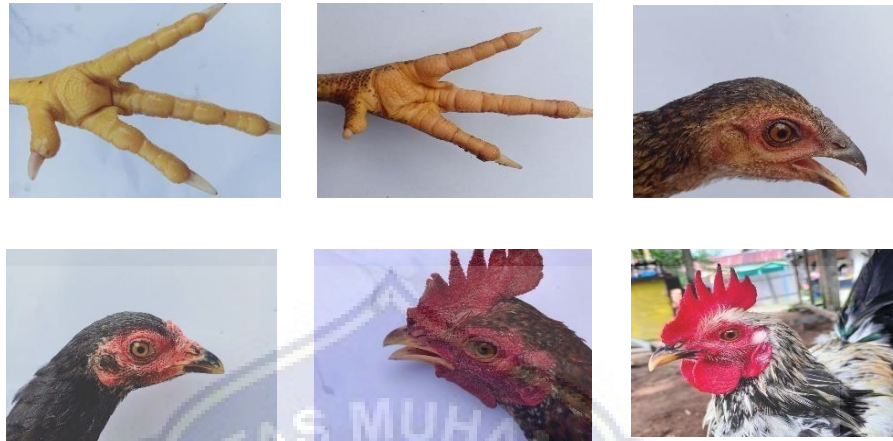
**Gambar 8.** Data Gambar dari Dinas Pertanian dan Ketahanan Pangan Kota Bulukumba

### 2. Lingkungan Kecamatan Bonto Tiro Kabupaten Bulukumba

Data yang didapatkan langsung dari Lingkungan Kecamatan Bonto Tiro Kabupaten Bulukumba merupakan data gambar area kaki dan mata dan jengger ayam yang sehat atau tidak terjangkit. Data gambar diambil secara langsung oleh penulis pada tanggal 03 juni 2023 s/d tanggal 20 juni



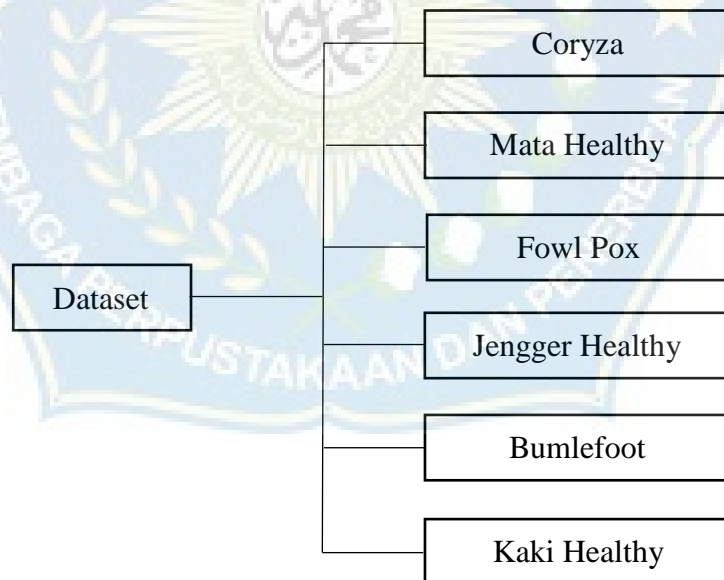
2023 menggunakan kamera *smartphone* dan data yang didapatkan adalah sebanyak 1.273 gambar.



**Gambar 9.** Data Gambar dari Lingkungan Kecamatan Bonto Tiro Kabupaten Bulukumba

### B. Pembagian *Dataset*

Sebelum dilakukan pembagian, *dataset* yang akan digunakan memiliki struktur direktori seperti berikut.



**Gambar 10.** Struktur Direktori *Dataset*

Gambar diatas merupakan struktur direktori dataset yang dimana terdapat dataset yang terbagi menjadi 6 kelas yaitu coryza, fowl pox, bumlefoot, mata healthy, kaki healthy, jengger healthy.

Pada setiap *folder*, jumlah *dataset* yang dimiliki berbeda-beda. Sebaran jumlah data citra pada tiap *folder* ditunjukkan pada

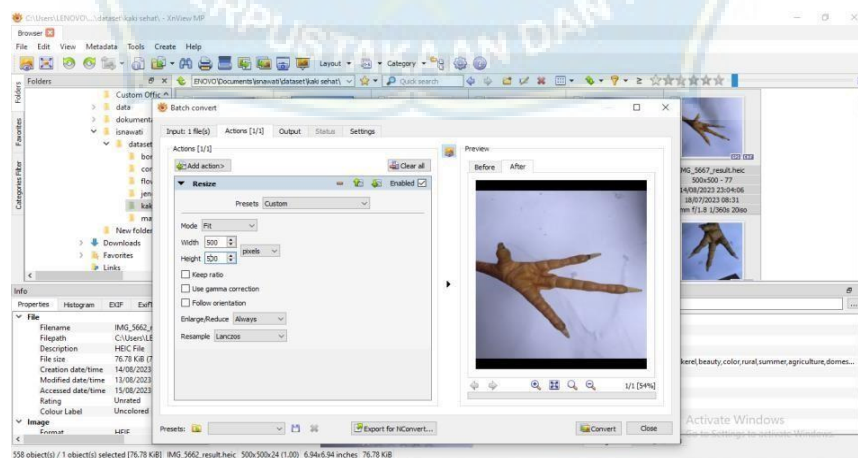
**Tabel 2.** Sebaran Jumlah Data Citra

No	Direktori	Jumlah Data
1.	Dataset/ penyakit ayam/bumlefoot	262
2.	Dataset/penyakit ayam/coryza	232
3.	Dataset/peyakit ayam/fowl pox	403
4.	Dataset/sehat/kaki healthy	234
5.	Dataset/sehat/mata healthy	560
6.	Dataset/sehat/jengger healthy	490
<b>Total</b>		<b>2.271</b>

### C. Resize

*Dataset* yang telah dibagi, kemudian diproses dengan melakukan *resize image*. Selain itu proses *resizing* bertujuan untuk menyamakan ukuran data citra.

Pada proses *resizing*, data citra akan di-resize menjadi ukuran 256x256 piksel. Setelah dilakukan proses *resizing* dengan jumlah data sebanyak 2.271 menggunakan aplikasi XnViewMP.



**Gambar 11.** Gambar untuk melakukan *resizing*

Gambar di atas merupakan proses *resizing* untuk menyamakan setiap ukuran data gambar menjadi 256x256, langkah pertama yaitu memilih folder yang akan di *resizing* lalu setelah itu pilih gambar dan kill tools pada bagian atas lalu pilih batch convert, klik actions lalu atur piksel gambar, dan klik output dan pilih folder untuk menyimpan gambar yang telah di *resizing*.

#### D. Training Data

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

Kode di atas merupakan kode untuk menghubungkan google drive dengan google colab.

```
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
```

kode di atas yaitu untuk mengimport beberapa modul seperti tensorflow dan keras untuk membangun model neural network. berikut merupakan penjelasan singkat fungsi yang di import:

1. sequential ini digunakan untuk membuat objek model sequential, yaitu model neural network dengan urutan lapisan yang saling terhubung secara berurutan.
2. BatchNormalizaion digunakan untuk menormalkan input pada setiap batch data selama proses pelatihan
3. Dense digunakan untuk meambah lapisan fully connected layer
4. Dropout digunakan sebagai teknik regulasi dalam penghindaran overfitting dengan secara acak mengabaikan sebagian node saat melatih model.
5. Flatten digunakan untuk meratakan output dari lapisan sebelumnya menjadi vector satu dimensi.
6. Conv2D dan MaxPooling2D, Convolutional layer (Conv2D) bertugas untuk ekstraksi fitur-fitur visual dari citra sedangkan max pooling layer

(MaxPooling2D) untuk melakukan reduksi ukuran spasial representasi fitur-fitur tersebut.

7. Activation digunakan untuk menambah fungsi aktivasi pada lapisan-lapisan neural network.

```
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
```

kode di atas yaitu untuk mengimport beberapa modul yang diperlukan tensorflow dan keras untuk pengolahan gambar dan optimasi diantaranya yaitu:

1. backend yang berfungsi untuk mengakses konfigurasi backend keras untuk menentukan urutan saluran.
2. image dan ImageData Generator. image digunakan untuk operasi dasar gambar sedangkan ImageDataGenerator digunakan untuk melakukan augmentasi data.
3. Adam dan SGD digunakan untuk memperbarui bobot-bobot pada model neural network saat proses pelatihan.

```
from tensorflow.keras.preprocessing.image import
img_to_array
from tensorflow.keras.callbacks import EarlyStopping from
tensorflow.keras.callbacks import ModelCheckpoint
```

kode ini diperlukan untuk pengolahan gambar dan pelatihan model neural network, berikut penjelasan singkat tentang fungsi-fungsi yang diimport:

1. img\_to\_array digunakan untuk mengubah objek gambar menjadi array numpy.
2. EarlyStopping digunakan sebagai callback (pemanggilan balik) untuk menghentikan proses pelatihan jika terjadi overfitting.
3. ModelCheckpoint merupakan callback yang digunakan untuk menyimpan bobot model secara periodic saat proses pelatihan berlangsung.

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

Kode di atas diperlukan sebagai operasi-operasi dasar pada data dan gambar, berikut penjelasan singkat mengenai fungsi yang di import:

1. `numpy as np` berfungsi melakukan operasi numerik pada array multidimensi
2. `matplotlib.pyplot as plt` berguna untuk visual grafik atau plot dari data, seperti histogram atau gambar.
3. `cv2` adalah pustaka OpenCV yang menyediakan berbagai fungsi serta algoritma pemrosesan citra computer
4. `pickle` digunakan untuk realisasi objek python menjadi format byte stream
5. `os.listdir` digunakan untuk mendapatkan daftar nama file dalam suatu direktori tertentu.

Kode ini melibatkan beberapa library yang umum digunakan dalam pembuatan model jaringan saraf, termasuk `sklearn`, `tensorflow`, `keras`, `numpy`, `matplotlib.pyplot`, dan `cv2`. Ada juga pemanggilan modul lainnya seperti `ImageDataGenerator` untuk augmentasi data gambar, serta callback seperti `EarlyStopping` dan `ModelCheckpoint`.

#### 1. *Inisialisasi Parameter Learning*

Tujuan dari perancangan serta melakukan pelatihan algoritma CNN ini adalah agar sistem dapat mengenali ciri dari setiap gambar dan kemudian menandai seluruh *neuron* serta menentukan dari setiap *neuron* tersebut yang mana yang akan diaktifkan ketika gambar diklasifikasi.

Sebelum dilakukan proses *training*, data *training* yang telah melalui proses augmentasi gambar perlu dipanggil terlebih dahulu dan perlu dilakukan inisialisasi terhadap beberapa parameter *learning*. Beberapa parameter *learning* yang perlu diinisialisasi untuk proses *training* ada tiga, yaitu *learning rate*, *batch size*, dan *epoch*.

```
EPOCHS = 100
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)
```

Pada bagian kode yang kamu berikan, terdapat beberapa variabel dan pengaturan yang digunakan dalam proses pelatihan model jaringan saraf. Berikut penjelasannya:

**Tabel 3. Inisialisasi Parameter**

Parameter	Deskripsi
<i>Learning rate</i>	Digunakan untuk menghitung nilai koreksi bobot pada saat proses training. Nilai learning rate yang diterapkan pada penelitian ini adalah 0,001 dengan <i>optimizer</i> Adam.
<i>Batch Size</i>	Digunakan untuk menentukan berapa jumlah sampel data yang disebar. Dalam penelitian ini nilai <i>batch size</i> yang digunakan adalah 32.
<i>Epoch</i>	Digunakan untuk penentu jumlah <i>epoch</i> yang harus dilakukan dalam proses <i>training dataset</i> . Jumlah <i>epoch</i> pada penelitian ini adalah 100.

Kemudian ada beberapa opsi optimizer:

- `opt = Adam(learning_rate=INIT_LR)`: Menggunakan optimizer Adam dengan learning rate sesuai dengan nilai `INIT_LR`.

Selain itu, dalam penelitian ini juga digunakan algoritma optimasi yaitu *Adam (Adaptive Moment Estimation) Optimizer*. Algoritma optimasi *Adam* ini merupakan salah satu algoritma optimasi yang banyak digunakan dalam *deep learning*. Beberapa penelitian mengenai *deep learning* telah menunjukkan bahwa dengan menggunakan *Adam* menghasilkan akurasi terbaik dibandingkan dengan algoritma lainnya (Bera & Shrivastava, 2020; Irfan et al., 2022; Wikarta et al., 2020). Kingma & Ba, 2014 memperkenalkan *Adam optimizer* pertama kali. Dalam *paper* tersebut, dapat disimpulkan beberapa keunggulan dari algoritma ini, yaitu efisien secara komputasi, persyaratan memori kecil, dan beberapa kelebihan lainnya.

```
mc = ModelCheckpoint(r'/content/Forest Fire CNN/ffs_cnn.h5',
monitor='val_accuracy', mode='max', verbose=1,
save_best_only=True)
```



```
# mc =
ModelCheckpoint(r'/content/drive/MyDrive/penyakit/ffs_cnn.h5',
, monitor='val_accuracy', verbose=1)
default_image_size = tuple((256, 256)) image_size = 0
directory_root =
r"/content/drive/MyDrive/Isna_2/Isna_2/penyakit" width=256
height=256 depth=3
```

Selanjutnya, terdapat juga callback:

- mc: ModelCheckpoint adalah sebuah callback untuk menyimpan bobot/model dengan akurasi validasi terbaik selama pelatihan. Model tersebut akan disimpan di path /content/Forest Fire CNN/ffs\_cnn.h5.

Lalu ada beberapa variabel lainnya seperti:

- default\_image\_size dan image\_size : Menentukan ukuran gambar input dalam tuple (width, height).
- directory\_root : Merupakan direktori root tempat dataset penyakit berada.
- Variabel-width, height, depth: Menentukan dimensi gambar (lebar, tinggi, dan kedalaman).

## 2. 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
```

Kode di atas sedang mengidentifikasi fungsi fit\_model yang menerima dua parameter yaitu n-nodes, dan idx yang bertujuan untuk membangun model arsitektur.

Kemudian membuat model objek menggunakan class Sequential() lalu mendefinisikan variable inputShape yang merupakan tuple berisi dimensi gambar. Variable chanDim digunakan untuk menentukan posisi saluran dalam data gambar. if\_else digunakan untuk memeriksa format data gambar apakah "channels\_first" atau bukan.

```
### 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))

```

Pada bagian ini ditambahkan beberapa lapisan ke dalam model diantaranya:

1. Conv2D adalah lapisan konvolusi dua dimensi yang digunakan untuk melakukan ekstraksi fitur dari gambar, (n\_nodes[idx][0]) adalah jumlah filter yang digunakan pada layer ini, (3,3) adalah ukuran filter yang digunakan. Kemudian padding="same" berfungsi mengatur agar output memiliki ukuran yang sama dengan input dan input\_shape mendefinisikan dimensi input.
2. Setelah itu ditambahkan fungsi aktivasi "relu" yang berfungsi memperkenalkan non-linearitas pada output dari lapisan sebelumnya.

3. Tanda pagar menandakan bahwa beberapa baris kode tersebut dijadikan komentar atau tidak dieksekusi.

```
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(6))
model.add(Activation("softmax"))
#####
#compile model & fit
```

Selanjutnya, terdapat lapisan Dense yang berfungsi untuk menambah *hidden layer* dengan *node* yang berjumlah 256 lagi dengan 6 node dan menggunakan activation function softmax. Pada proses *dropout* ini diberikan probabilitas bernilai 0,5 pada setiap *neuron* dan proses ini akan mencegah *overfitting* sekaligus mempercepat proses pembelajaran.

Kemudian ditambahkan beberapa lapisan lain diantaranya;

1. Flatten() berfungsi untuk meratakan output dari lapisan sebelumnya menjadi satu dimensi
2. lalu ditambahkan layer Dense dengan jumlah neuron sesuai dengan `n_nodes[idx][0]`. Layer Dense ini berfungsi menghubungkan setiap neuron pada lapisan sebelumnya.

```
# model.compile(loss="binary_crossentropy",
optimizer=opt,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
)
```

Pada kode di atas dilakukan komplikasi dan pelatihan model.

1. `.compile()` digunakan untuk mengkonfigurasi proses pelatihan, `loss='categorical_crossentropy'` adalah fungsi loss yang digunakan untuk mengukur seberapa baik performa model. Kemudian `metrics=["accuracy"]` berfungsi menentukan metric evaluasi yang digunakan selama pelatihan.
2. Lalu di cetak pesan "training st Arch!" yang berfungsi memberikan informasi mengenai arsitektur mana yang sedang di bor.
3. setelah itu dilakukan pemanggilan fungsi `.fit()` untuk melatih model. `(aug.flow())` digunakan untuk membuat augmented dataset dengan ukuran batch tertentu dari data latih (`x_train` dan `y_train`), (`validation_data=(x_test, y_test)`), kemudian parameter lain seperti jumlah langkah epoch (`steps_per_epoch`), jumlah epoch (`epoch`), tingkat verbositas (`verbose`), dan `[mc]` merupakan `ModelCheckpoint`.

```
#####
#plot learning curves
acc = history.history['accuracy']
epoch = range(1, len(acc) + 1)
```

Bagian kode di atas digunakan untuk membuat plot kurva pembelajaran dari data pelatihan model dan terdapat beberapa bagian-bagian dan berikut adalah penjelasannya:

1. `acc = history.history['accuracy']` berfungsi menyimpan nilai akurasi yang diperoleh dari objek `hstory`.
2. `epoch = range(1, len(acc)+ 1)` berisi rentang nilai epoch.

```
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()
```

kode di atas digunakan untuk membuat subplot pada gravik kurva.

1. `plt.subplot(len(n_nodes), 2, (idx+1)*2-1)`, fungsi `subplot()` diguakan untuk membagi area gambar menjadi beberapa subplot dengan jumlah baris dan

kolom tertentu. Argumen pertama (`len(n_nodes)`) menunjukkan jumlah baris pada grid subplot, argumen kedua (`2`) menunjukkan jumlah kolom, `((idx+1)*2-1)` menunjukkan nomor urutan dari subplot.

2. `plt.plot(epoch, history.history['accuracy'], label='train acc')` kode ini memplot garis kurva akurasi pelatihan model dengan menggunakan data dari `history`.
3. `plt.title(f'Accuracy, node {n_nodes[idx]}')` berfungsi memberikan judul pada plot.
4. `plt.legend()` kode ini menambahkan legenda pada plot.

Setelah proses *dropout*, dilakukan inisialisasi *layer output* yang berupa 6 *node* karena dalam pengkalsifikasian di penelitian ini terdapat 6 kelas, yaitu 'bumlefoot', 'coryza', 'fowl pox', 'mata healthy', 'kaki healthy', 'jengger healthy'. Karena lebih dari dua kelas atau *multi-class classification*, maka akan digunakan fungsi aktivasi *softmax classifier*. Fungsi aktivasi *softmax* akan mengembalikan probabilitas dari masing-masing kelas dan kelas target akan memiliki nilai probabilitas lebih tinggi dibandingkan dengan kelas yang lain.

Setelah menginisialisasi model, langkah berikutnya adalah melakukan kompilasi dan pelatihan pada data latih. Model akan dioptimalkan dengan menggunakan *categorical cross-entropy loss function*. Selama proses pelatihan, akan ada juga evaluasi performa pada data validasi.

Dengan demikian, kita bisa memvisualisasikan perubahan akurasi dan loss selama proses pelatihan model secara lebih jelas melalui grafik-garafik tersebut.

#### E. Hasil Proses *Training*

```
n_nodes = [  
    256, 128, 64, 32  
]  
plt.figure(figsize=(13, 4*len(n_nodes)))  
for x in range(len(n_nodes)):
```

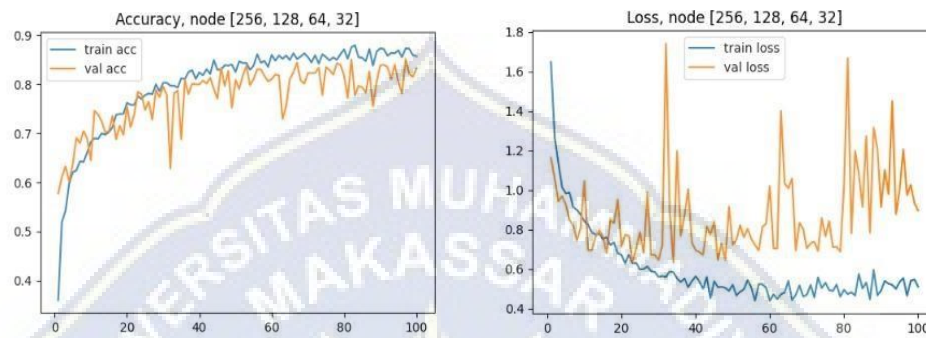
```

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

# show learning curves plt.show()

```

Sesuai dengan skenario yang telah dijelaskan sebelumnya, pada bagian ini dilakukan percobaan dengan membagi seluruh data dengan data *training* sebesar 70%, data *validation* 20%, dan data *testing* 10%.



**Gambar 12.** Visualisasi Akurasi *Training* dan *Validation Training Accuracy*

Akurasi dari *training* dengan 100 *epochs* mengalami peningkatan pada setiap *epoch*. Dari proses *training* yang dilakukan, dapat disimpulkan bahwa rata-rata akurasi dari *train* adalah sebesar 83%. Akurasi tertinggi yang diperoleh terjadi pada *epoch* ke-100, yaitu sebesar 85% dan akurasi terendah terjadi pada *epoch* pertama, yaitu sebesar 37%.

#### 1. *Validation Accuracy*

Akurasi dari validasi dengan 100 *epochs* mengalami naik turun bahkan mengalami *overfit* pada setiap *epoch* di 40 *epochs* pertama. Setelah *epoch* ke 40, akurasi pada setiap *epoch* hampir selalu konsisten hingga *epoch* terakhir. Akurasi tertinggi yang didapatkan terdapat pada *epoch* ke-97, yaitu sebesar 87%.

#### 2. *Loss*

Nilai dari *loss* pada *training* mengalami penurunan hingga *epoch* terakhir. Nilai *training loss* tertinggi terjadi pada *epoch* pertama, yaitu 1,6 dan terendah pada *epoch* 94, yaitu 0,49. Nilai akurasi pada *validation loss* mengalami naik turun hingga *epoch* ke-50 dan perlahan mengalami penurunan nilai *loss* hingga *epoch* terakhir.

```
from keras.models import load_model
```

Fungsi `from keras.models import load_model` digunakan untuk mengimpor modul `load_model` dari pustaka Keras. Modul ini berguna untuk memuat (load) model yang telah disimpan sebelumnya dalam format file.

Setelah kita memuat modul tersebut, kita dapat menggunakan fungsi `load_model()` untuk memuat model yang ingin kita gunakan. Model tersebut harus disimpan terlebih dahulu dalam format file seperti `.h5` atau `.hdf5`.

```
fs_7030 = load_model('/content/Forest FireCNN/ffs_cnn.h5')
```

Fungsi `load_model()` untuk memuat model yang disimpan dalam file dengan nama `/content/Forest Fire CNN/ffs_cnn.h5`. Fungsi tersebut akan mengembalikan objek model yang telah dimuat dari file tersebut.

```
loss, acc = fs_7030.evaluate(x_train, y_train)
```

Baris kode tersebut digunakan untuk mengevaluasi model menggunakan data latih (`x_train` dan `y_train`). Hasil evaluasi berupa `loss` (kerugian) dan `accuracy` (akurasi) dari model pada data latih akan disimpan dalam variabel `loss` dan `acc`.

## **F. Load And Testing**

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

Fungsinya adalah untuk mengimpor fungsi `load_model` dari modul `tensorflow.keras.models`. Fungsi ini digunakan untuk memuat (load) model yang telah disimpan sebelumnya dalam format file. Dengan menggunakan fungsi ini, kita dapat dengan mudah mengambil kembali model yang sudah dilatih dan digunakan untuk keperluan prediksi atau evaluasi.

Misalnya, jika kita telah melatih sebuah model dan menyimpannya sebagai file `.h5`, maka kita dapat menggunakan `load_model` untuk memuat kembali model tersebut ke dalam variabel Python. Setelah dimuat, kita bisa langsung menggunakan model tersebut tanpa perlu melatih ulang.

```
cs_adam_10 = load_model('/content/Forest FireCNN/ffs_cnn.h5')
```



Kode ini untuk memuat model yang telah disimpan dalam file 'ffs\_cnn.h5' menggunakan fungsi load\_model() dari library Keras.

```
gmb1 =  
cv2.imread('/content/drive/MyDrive/Isna_2/Isna_2/penyakit/penyakit/mata healthy/IMG20230603063126_result_result.jpg')  
image_rgb = cv2.cvtColor(gmb1, cv2.COLOR_BGR2RGB)  
plt.imshow(image_rgb)
```

Kode di atas adalah untuk mengubah format warna gambar dari BGR (BlueGreen-Red) menjadi RGB (Red-Green-Blue). Setelah itu, gambar tersebut ditampilkan menggunakan matplotlib.pyplot.imshow(). gmb1 merupakan variabel yang berisi gambar dalam format BGR. Dengan menggunakan cv2.cvtColor() dan parameter cv2.COLOR\_BGR2RGB, kita dapat mengonversi gambar tersebut ke format RGB.

Setelah konversi selesai, hasilnya disimpan dalam variabel image\_rgb. Kemudian, dengan menggunakan plt.imshow(image\_rgb), kita dapat menampilkan gambar tersebut dalam notebook atau aplikasi Python lainnya.

```
# Prepare image test  
  
# img_test =  
cv2.imread('/content/drive/MyDrive/Isna_2/Isna_2/penyakit/penyakit/coryza/shs1_result_result.jpg') img3 =  
cv2.resize(gmb1, default_image_size) img3 =  
img_to_array(img3) img3 = np.expand_dims(img3, axis=0) img3  
= img3/225
```

Bagian ini adalah persiapan gambar uji sebelum digunakan pada model. Pertama, gambar diresize menggunakan fungsi cv2.resize dengan ukuran default yang telah ditentukan sebelumnya. Kemudian, gambar tersebut diubah menjadi array menggunakan fungsi img\_to\_array. Selanjutnya, dimensi array gambar diperluas dengan menambahkan satu dimensi baru menggunakan np.expand\_dims, agar sesuai dengan input yang diterima oleh model.

Terakhir, nilai pixel dalam array gambar dinormalisasi dengan membaginya dengan 225. Hal ini dilakukan untuk mengubah rentang nilai piksel menjadi antara 0 dan 1.



```
img3.shape y_pred = cs_adam_10.predict(img3)
y_pred
```

Baris kode tersebut menunjukkan penggunaan model `cs_adam_10` untuk melakukan prediksi pada data gambar `img3`. Prediksi dilakukan dengan memanggil metode `.predict()` dari objek model, dan hasilnya akan disimpan dalam variabel `y_pred`.

```
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
```

## G. Pengaruh *Hyperparameters* terhadap Akurasi

Proses *training* pada jaringan diatur dengan mendefinisikan beberapa variabel yang disebut *hyperparameter*, seperti *learning rate*, jumlah *hidden layer*, jumlah *neuron*, fungsi aktivasi, jumlah *training epoch* dan lainnya (Atteia et al., 2022). *Hyperparameter* sangat memberikan dampak pada pelatihan serta kinerja model. Oleh karena itu, menetapkan *hyperparameter* terbaik merupakan langkah yang penting dalam membuat model yang akan menghasilkan akurasi dan prediksi yang baik.

### 1. Jumlah *Training Epoch*

*Epoch* merupakan sebuah siklus algoritma *learning* dari seluruh data *training*. Satu *epoch* menandakan sebuah algoritma *machine learning* yang telah ‘mempelajari’ data *training* secara keseluruhan (Wibawa, 2017). Wibawa mengatakan bahwa nilai *epoch* yang sesuai tidak dapat diketahui dan hingga saat ini belum ada penelitian yang melakukan klaim mengenai jumlah *epoch* terbaik dalam melakukan proses pembelajaran pada data *training*. Pada penelitian ini akan diujikan beberapa nilai *epoch* agar dapat diketahui nilai akurasi *training* yang terbaik.

Pada penelitian ini akan dilakukan pengujian perbandingan antara fungsi aktivasi, rule-softmax dan softmax-rule

**Table 4.** pengaruh Jumlah *Epoch* (Percobaan rule-softmax)

<i>Epoch</i>	Akurasi <i>Training</i>	Akurasi <i>Validation</i>	Akurasi <i>Testing</i>	<i>Loss</i> <i>Value</i>	Waktu
50	92%	83%	90%	0,01	20 menit
80	92%	84%	99%	0,02	26 menit
100	91%	84%	100%	0,01	28 menit
150	92%	85%	100%	0,01	32 menit

Berdasarkan tabel di atas, dengan menggunakan nilai *learning rate* 0,001, dimensi citra 256x256, dan nilai *batch size* 8, didapatkan akurasi yang paling optimal adalah dengan jumlah *epoch* 100. Dari tabel di atas terdapat tiga kali percobaan dengan *epoch*, akurasi, *loss value*, dan waktu yang berbeda. Pada *epoch* 50 akurasi *training* sebesar 92%, akurasi *validation* sebesar 83%, akurasi *testing* 90%, *loss value* 0,01, dan waktu 20 menit. Pada *epoch* 80 jumlah akurasi *training* 92%, akurasi *validation* 84%, akurasi *testing* 99%, *loss value* 0.02, dan waktu 26 menit. Pada *epoch* 100 akurasi *training* 91%, akurasi *validation* 84%, akurasi *testing* 100%, *loss value* 0,01, dan waktu 28 menit. Pada *epoch* 150 jumlah akurasi *training* 92%, akurasi *validation* 85%, akurasi *testing* 100%, *loss value* 0,01 dan waktu 32 menit. Jumlah *epoch* 100 dan 150 memiliki nilai akurasi *training* dan nilai *loss* yang hampir sama tetapi terdapat perbedaan pada nilai akurasi *validation* dan *testing* sehingga dapat disimpulkan bahwa akurasi terbaik didapatkan ketika model menerapkan jumlah *epoch* sebanyak 150 pada saat proses *training*.

**Table 5.** pengaruh Jumlah *Epoch* (Percobaan softmax-relu)

<i>Epoch</i>	Akurasi <i>Training</i>	Akurasi <i>Validation</i>	Akurasi <i>Testing</i>	<i>Loss</i> <i>Value</i>	Waktu
50	12%	15%	20%	13,1	20 menit
80	18%	18%	26%	8,4	26 menit
100	25%	27%	29%	6,1	28 menit
150	26%	28%	30%	5,9	32 menit

Berdasarkan tabel di atas, dengan menggunakan nilai *learning rate* 0,001, dimensi citra 256x256, dan nilai *batch size* 32, didapatkan akurasi yang paling

optimal adalah dengan jumlah *epoch* 150. Dari tabel di atas terdapat empat kali percobaan dengan *epoch*, akurasi, *loss value*, dan waktu yang berbeda. Pada *epoch* 50 akurasi *training* sebesar 12%, akurasi *validation* sebesar 15%, akurasi *testing* 20%, *loss value* 13,1, dan waktu 20 menit. Pada *epoch* 80 jumlah akurasi *training* 18%, akurasi *validation* 18%, akurasi *testing* 26%, *loss value* 8,4, dan waktu 26 menit. Pada *epoch* 100 akurasi *training* 25%, akurasi *validation* 27%, akurasi *testing* 29%, *loss value* 6,1, dan waktu 28 menit. Pada *epoch* 150 jumlah akurasi *training* sebesar 26%, akurasi *validation* sebesar 28%, akurasi *testing* 30%, *loss value* 5,9, dan waktu 32 menit. Jumlah *epoch* 100 dan 150 memiliki nilai akurasi *training* dan nilai *loss* yang hampir sama tetapi terdapat perbedaan pada nilai akurasi *validation* dan *testing* sehingga dapat disimpulkan bahwa akurasi terbaik belum didapatkan bahkan ketika model menerapkan jumlah *epoch* sebanyak 150 pada saat proses *training* namun nilai akurasi sangat rendah dan nilai *loss* tinggi.

Dapat disimpulkan bahwa kedua perbandingan fungsi aktivasi di atas yang paling bagus dalam menentukan akurasi adalah relu-softmax dibandingkan softmax-relu

## H. Hasil Proses Testing

Setelah arsitektur model ditentukan, dilakukan proses *training*, hingga mendapatkan arsitektur dengan *hyperparameter* terbaik, kemudian dilanjutkan dengan pengujian model. Pengujian ini menggunakan data uji sebanyak 12 citra dengan jumlah setiap kelas berbeda-beda. Seperti yang telah dijelaskan sebelumnya, jumlah data uji pada tiap kelas ini didapatkan dari hasil pembagian data *training*, *validation*, dan *testing*. Sebaran data uji pada tiap kelas ditunjukkan pada tabel 9.

**Tabel 6.** Jumlah Data Uji pada Tiap Kelas

No	Direktori	Jumlah Data
1.	Dataset/ penyakit ayam/bumlefoot	2
2.	Dataset/penyakit ayam/coryza	2
3.	Dataset/peyakit ayam/fowl pox	2
4.	Dataset/sehat/kaki healthy	1

5.	Dataset/sehat/mata healthy	2
6.	Dataset/sehat/jengger healthy	2
Total		11

Tabel di atas merupakan tabel data uji dari setiap kelas dan memiliki jumlah data yang berbeda di setiap kelasnya.



**Gambar 13.** Sampel Data *Testing*

Dari gambar di atas adalah beberapa gambar yang telah melakukan percobaan di luar dataset yang ada pada sistem dengan piksel 256x256, jumlah epoch yang digunakan 100, batch size 8 yang menghasilkan akurasi testing sebesar 94%.

## **BAB V**

### **KESIMPULAN DAN SARAN**

#### **A. Kesimpulan**

Berdasarkan penelitian yang telah dikerjakan, maka dapat ditarik beberapa kesimpulan sebagai berikut:

1. Dari hasil proses konfigurasi *hyperparameter*, didapatkan akurasi terbaik dengan *epoch* berjumlah 100 dan *batch size* 32. Dengan skenario perbandingan *dataset train* sebesar 70%, *dataset validation* 20%, dan *dataset test* sebesar 10%, digunakan *Input* citra yang berdimensi 256x256 piksel dan ukuran *learning rate* 0,001.
2. Hasil dari akurasi *testing* yang didapatkan dengan arsitektur yang telah dibuat adalah sebesar 95% yang mana dari 11 data *test* yang diberikan, 9 di antaranya terprediksi dengan benar dan prediksi tersebut sesuai dengan kelas yang sebenarnya. Dari hasil *recall*, *precision*, dan *f1-score* pada kelas ‘kaki healthy dan ‘bumlefoot’ menghasilkan akurasi yang lebih rendah dibandingkan kelas lainnya dan ini disebabkan karena beberapa data gambar kaki ayam dalam keadaan tidak bersih (terdapat kotoran seperti tanah/debu yang menutupi sebagian luka) sehingga sistem kesulitan mempelajari setiap piksel pada gambar yang terdapat luka lepuh.
3. Hasil dari penelitian ini menunjukkan bahwa fungsi aktivasi yang paling bagus dalam mempercepat proses perubahan model dan akurasi adalah dengan menggunakan *relu-softmax* dalam mendeteksi penyakit pada ayam.

#### **B. Saran**

Berdasarkan penelitian yang telah dilakukan, beberapa saran yang penulis berikan sebagai upaya perbaikan untuk penelitian selanjutnya adalah sebagai berikut:

1. Menambahkan beberapa *hyperparameter* sebagai pembanding untuk memperoleh arsitektur CNN yang menghasilkan akurasi lebih baik lagi.
2. Sektor peternakan dan kesehatan hewan di Indonesia perlu didorong untuk selalu melakukan kegiatan dokumentasi terhadap gejala awal pada hewan yang terkena penyakit coeyza, fowl pox, dan bumlefoot. Hal ini diperlukan untuk mempermudah penulis dalam memperoleh dan memperbanyak jumlah *dataset* yang dapat dipelajari oleh model sehingga pendeteksian dapat semakin akurat.



## DAFTAR PUSTAKA

- Ariantika, T. D. S., Ramadhan, M., & Santoso, I. (2022). Sistem Pakar Mendiagnosa Penyakit Fowlpox (Cacar Unggas) Pada Burung Puyuh Menggunakan Metode Teorema Bayes. *Jurnal Cyber Tech*, 4(1).
- A'YUN, ZAYYINATUL. SISTEM PAKAR DIAGNOSA PENYAKIT PADA AYAM PETELUR DENGAN MENGGUNAKAN METODE FORWARD CHAINING. 2014. PhD Thesis. Universitas Muhammadiyah Ponorogo.
- Azhari, R., Nugroho, N. B., & Zunaidi, M. (2020). Sistem Pakar Dalam Mendiagnosa Penyakit Tetelo (Newcastle disease/ND) Pada Ayam Jago Dengan Menggunakan Metode Theorema Bayes. *Jurnal Cyber Tech*, 3(8).
- Bere, J., Irawan, J. D., & Ariwibisono, F. X. (2021). Sistem Pakar Diagnosis Penyakit Pada Ayam Menggunakan Metode Certainty Factor. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 5(1), 217-224.
- FAIZAH, A. N. (2022). IMPLEMENTASI CERTAINTY FACTOR DALAM DIAGNOSA PENYAKIT PADA AYAM BROILER (Doctoral dissertation, Universitas Nahdlatul Ulama Sunan Giri).
- Hayadi, B. H. (2018). *Sistem pakar*. Deepublish.
- Kholik, A. (2021). Klasifikasi Menggunakan Convolutional Neural Network (Cnn) Pada Tangkapan Layar Halaman Instagram. *Jurnal Data Mining dan Sistem Informasi*, 2(2), 10-20.
- La Ode Ansyarullah, S. S. Klasifikasi Cats dan Dogs dengan Metode CNN dalam Fungsi Aktivasi relu, sigmoid, softmax, softplus, softsign, dan selu.
- Muafi, M., Wijaya, A., & Aziz, V. A. (2020). Sistem Pakar Mendiagnosa Penyakit Mata Pada Manusia Menggunakan Metode Forward Chaining. *COREAI: Jurnal Kecerdasan Buatan, Komputasi dan Teknologi Informasi*, 1(1), 43-49.
- Muslimin, D. B., Kusmanto, D., Amilia, K. F., Ariffin, M. S., Mardiana, S., & Yulianti, Y. (2020). Pengujian Black Box pada Aplikasi Sistem Informasi Akademik Menggunakan Teknik Equivalence Partitioning. *Jurnal Informatika Universitas Pamulang*, 5(1), 19-25.



- Nansia, O., & Sinag, B. (2019). Sistem Pakar Mendiagnosa Penyakit Pada Ayam Ternak Menggunakan Metode Certainty Faktor. *Journal Of Informatic Pelita Nusantara*, 4(2), 14-18.
- Nurkholiq, A., Saryono, O., & Setiawan, I. (2019). Analisis pengendalian kualitas (quality control) dalam meningkatkan kualitas produk. *Jurnal Ekonologi Ilmu Manajemen*, 6(2), 393-399.
- Putra, Y. F. (2020). **APLIKASI SISTEM PAKAR DIAGNOSA PENYAKIT PADA PERNAFASAN AYAM BROILER MENGGUNAKAN METODE FORWARD CHAINING BERBASIS WEBSITE** (Doctoral dissertation, Universitas Muhammadiyah Sukabumi).
- SETIAWAN, Budi. **PENERAPAN ALGORITMA YOU ONLY LOOK ONCE (YOLO) UNTUK DETEKSI TANAMAN MIANA BERBASIS ANDROID**. 2021. PhD Thesis. Universitas Muhammadiyah Ponorogo.
- Simanungkalit, I. R., & Sinaga, B. (2021). Sistem Pakar Mendiagnosa Penyakit Ayam Kampung Dengan Menggunakan Metode Certainty Factor Berbasis Web. *Jurnal Nasional Komputasi dan Teknologi Informasi*, 4(6), 390-396.
- Widodo, A. S., WULANNINGRUM, R., & SWANJAYA, D. (2022). Sistem Pakar Penyakit Daun Kacang Tanah Menggunakan Metode CNN (Convolutional Neural Network) (Doctoral dissertation, Universitas Nusantara PGRI Kediri).
- Zebua, H. (2023). **PEMBUATAN MODEL DETEKSI GEJALA AWAL PENYAKIT MULUT DAN KUKU PADA SAPI BERBASIS CITRA MENGGUNAKAN METODE CONVOLUTIONAL NEURAL NETWORK** (Doctoral dissertation, Sistem Informasi).



Lampiran 1. Surat Izin Penelitian





**PEMERINTAH KABUPATEN BULUKUMBA  
DINAS PENANAMAN MODAL, PELAYANAN TERPADU  
SATU PINTU DAN TENAGA KERJA  
( D P M P T S P T K )**

Jl. Kenari No. 13 Telp. (0413) 84241 Fax. (0413) 85060 Bulukumba 92511

**SURAT IZIN PENELITIAN  
NOMOR : 435/DPMTSPTK/IP/VII/2023**

Berdasarkan Surat Rekomendasi Teknis dari KESBANGPOL dengan Nomor 074/808/Bakesbangpol/VII/2023 tanggal 26 Juli 2023, Perihal Rekomendasi Izin Penelitian maka yang tersebut dibawah ini :

Nama Lengkap	: Isnawati
Nomor Pokok	: 105841110319
Program Studi	: Teknik Informatika
Jenjang	: S1
Institusi	: Universitas Muhammadiyah Makassar
Tempat/Tanggal Lahir	: Usa / 2001-09-21
Alamat	: Dusun Usa
Jenis Penelitian	: Kualitatif
Judul Penelitian	: SISITEM PAKAR DETEKSI CITRA PADA PENYAKIT AYAM DI KECAMATAN BONTO TIRO KABUPATEN BULUKUMBA
Lokasi Penelitian	: Bulukumba
Pendamping	: (1). Titin Wahyuni, S.Pd.,M.T. (2). Fahrin Irahman Rachman S.Kom.,MT
Instansi Penelitian	: Kecamatan Bonto Tiro Kabupaten Bulukumba
Lama Penelitian	: tanggal 3 Juli 2023 s/d 20 Juli 2023

Sehubungan dengan hal tersebut di atas, pada prinsipnya kami mengizinkan yang bersangkutan untuk melaksanakan kegiatan tersebut dengan ketentuan sebagai berikut :

1. Mematuhi semua Peraturan Perundang - Undangan yang berlaku dan mengindahkan adat - istiadat yang berlaku pada masyarakat setempat;
2. Tidak mengganggu keamanan/ketertiban masyarakat setempat
3. Melaporkan hasil pelaksanaan penelitian/pengambilan data serta menyerahkan 1(satu) eksamplar hasilnya kepada Bupati Bulukumba Cq. Kepala Badan Kesatuan Bangsa dan Politik Kab.Bulukumba;
4. Surat izin ini akan dicabut atau dianggap tidak berlaku apabila yang bersangkutan tidak memenuhi ketentuan sebagaimana tersebut di atas, atau sampai dengan batas waktu yang telah ditentukan kegiatan penelitian/pengumpulan data dimaksud belum selesai.

Dikeluarkan di : Bulukumba  
Pada Tanggal : 26 Juli 2023



Kepala Dinas DPMTSPTK

Dra. Hj. Umrah Aswani, MM  
Pangkat : Pembina Utama Muda-IV/c  
Nip : 19670304 199303 2 010



Balai  
Sertifikasi  
Elektronik

Dokumen ini telah ditandatangani secara elektronik menggunakan sertifikat elektronik yang diterbitkan oleh Balai Sertifikasi Elektronik (BSrE), BSSN



**PEMERINTAH KABUPATEN BULUKUMBA  
DINAS PERTANIAN DAN KETAHANAN PANGAN**

Jalan Sultan Hasanuddin No. 44 Telp/Fax (0413) 2510128, Bulukumba 92514  
E-mail : [diphp.blk@gmail.com](mailto:diphp.blk@gmail.com), Website : [www.diphbu.bulukumba.kab.go.id](http://www.diphbu.bulukumba.kab.go.id)

**SURAT KETERANGAN PENELITIAN**

Nomor : 620 / 570 / DPKP-BLK / VII / 2022

Bertanda tangan di bawah ini Kepala Dinas Pertanian dan Ketahanan Pangan Kab. Bulukumba, dengan ini menerangkan bahwa :

NAMA : ISNAWATI  
NIM : 105841110319  
FAKULTAS : TEKNIK  
PROGRAM STUDI : INFORMATIKA  
LOKASI PENELITIAN : KECAMATAN BONTO TIRO  
KABUPATEN BULUKUMBA

Yang bersangkutan telah melakukan Penelitian di Kecamatan Bonto Tiro Kab. Bulukumba Provinsi Sulawesi Selatan, terhitung mulai Tanggal 3 Juli 2023 s.d 20 Juli 2023 guna penulisan Skripsi dengan judul "**SISTEM PAKAR MENDIAGNOSA PENYAKIT PADA AYAM MENGGUNAKAN DETEKSI CITRA DI KECAMATAN BONTO TIRO KABUPATEN BULUKUMBA**".

Demikian Surat Keterangan penelitian ini dibuat untuk dapat dipergunakan sebagaimana mestinya.

Bulukumba 27 Juli 2023

KEPALA DINAS M.

**Dr. MUH. THAIYEB MANINGKASI, M. Si**

Pangkat : Pembina Tk. 1

NIP : 19680622 198903 1 007

Tembusan Kepada Yth :

1. Kepala Dinas Pertanian dan Ketahanan Pangan Kab. Bulukumba
2. Yang Bersangkutan
3. Arsip

```

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

# !apt-get install unrar

# !unrar x "/content/drive/MyDrive/Isna_1/Isna_2/penyakit"
"/content/drive/MyDrive/Isna_1/Isna_2/penyakit"

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 = 100
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'/content/drive/MyDrive/penyakit/ffs_cnn.h5',
monitor='val_accuracy', verbose=1)

default_image_size = tuple((256, 256))
image_size = 0
directory_root =
r"/content/drive/MyDrive/Isna_1/Isna_2/penyakit"
width=256
height=256
depth=3

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

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_folde
r}/")

        for single_plant_disease_image in
plant_disease_image_list :
            if single_plant_disease_image == ".DS_Store" :
                plant_disease_image_list.remove(single_plant_d
isease_image)

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

                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}")

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_)

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

x_train, x_test, y_train, y_test =
train_test_split(image_list, image_labels, test_size=0.3,
random_state=42)

import gc
gc.collect()

```

```

del(image_list)
del(image_labels)

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()

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(6))
model.add(Activation("softmax"))
#model.add(Activation("sigmoid"))

#####
#compile model & fit

# model.compile(loss="binary_crossentropy",
optimizer=opt,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()

import gc

gc.collect()

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()

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)

loss, acc = fs_7030.evaluate(x_test, y_test)

from tensorflow.keras.models import load_model

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

gmbr1 = cv2.imread('/content/drive/MyDrive/data
tess/bomlefoot/WhatsApp Image 2023-08-21 at
21.44.03_result.jpg')

image_rgb = cv2.cvtColor(gmbr1, cv2.COLOR_BGR2RGB)
plt.imshow(image_rgb)

```

```

# Prepare image test

# img_test = cv2.imread('/content/drive/MyDrive/data
testing/WhatsApp Image 2023-08-17 at 17.50.24.jpeg')
img3 = cv2.resize(gmbr1, default_image_size)
img3 = img_to_array(img3)
img3 = np.expand_dims(img3, axis=0)
img3 = img3/225

img3.shape

y_pred = cs_adam_10.predict(img3)

y_pred

# 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)

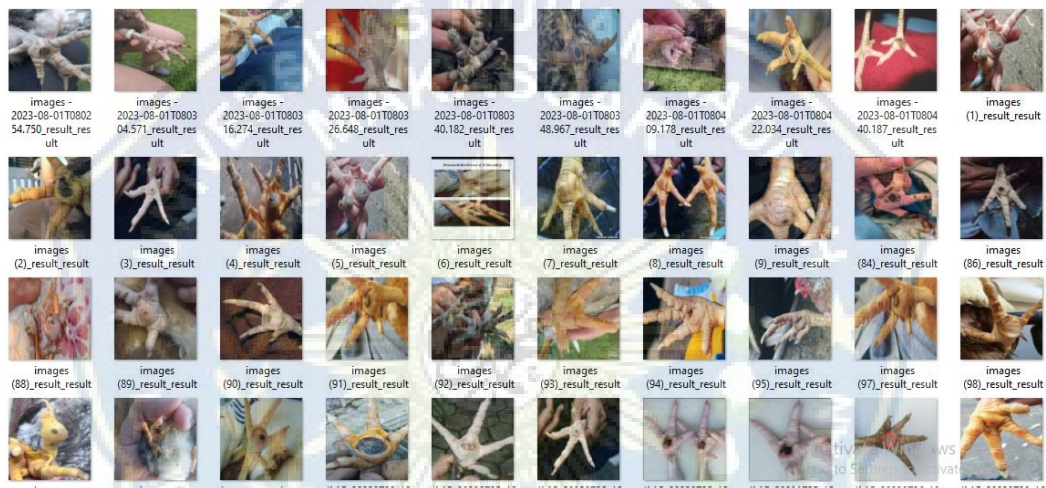
y_invers

```

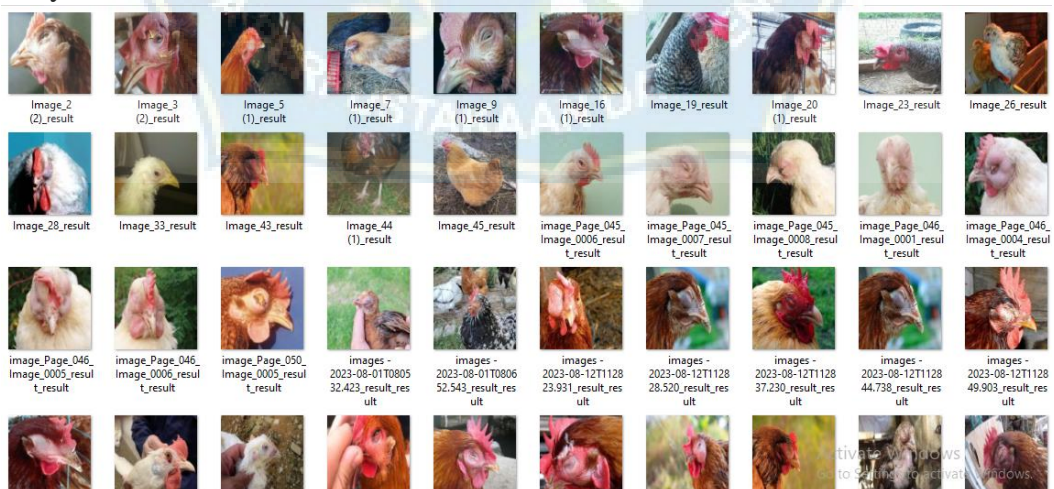
## Lampiran 4. Dataset

```
[INFO] Loading images ...  
[INFO] Processing mata healthy ...  
[INFO] Processing fowl pox ...  
[INFO] Processing bumlefoot ...  
[INFO] Processing coryza ...  
[INFO] Processing jengger healthy ...  
[INFO] Processing Bukan Data Ayam ...  
[INFO] Processing kaki healthy ...  
[INFO] Image loading completed
```

### Bumlefoot

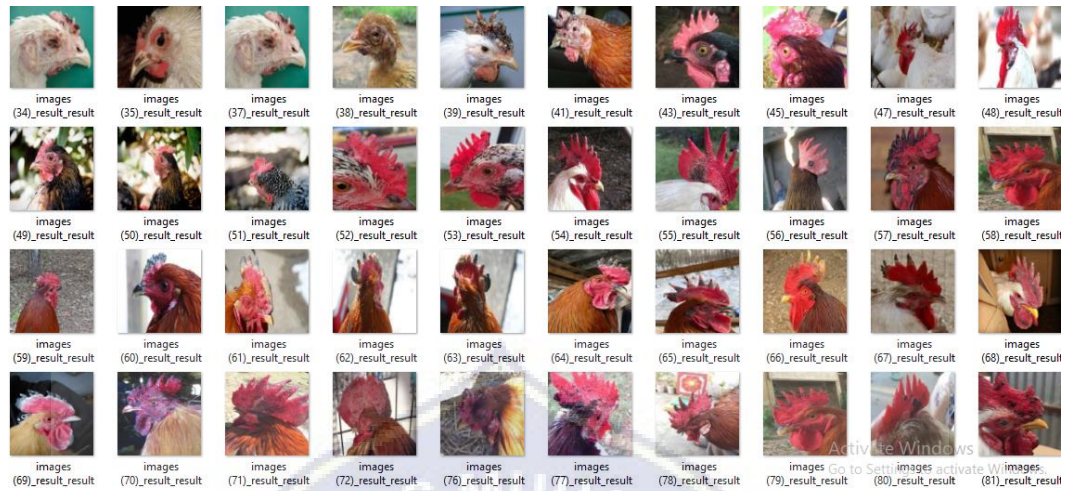


### Coryza





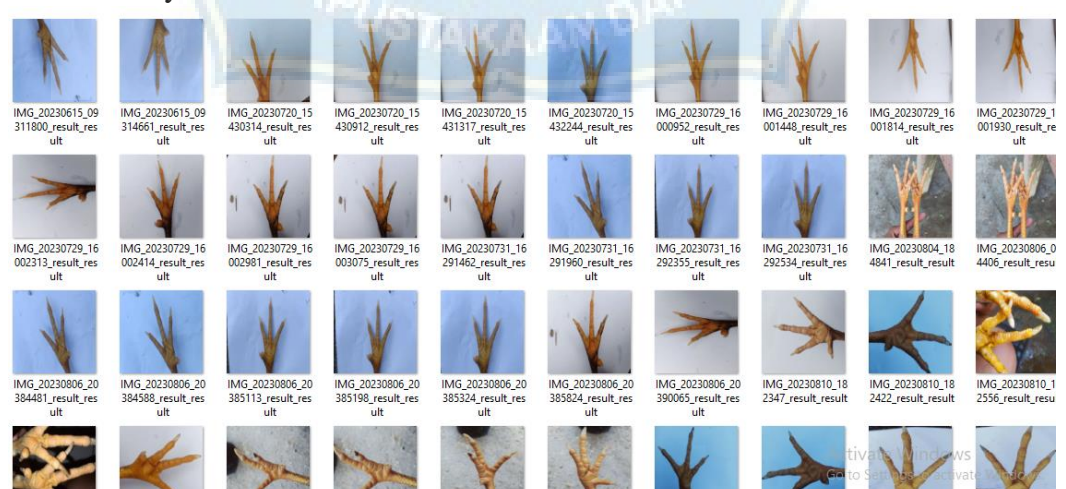
## Fowl Pox



## Jengger Healthy

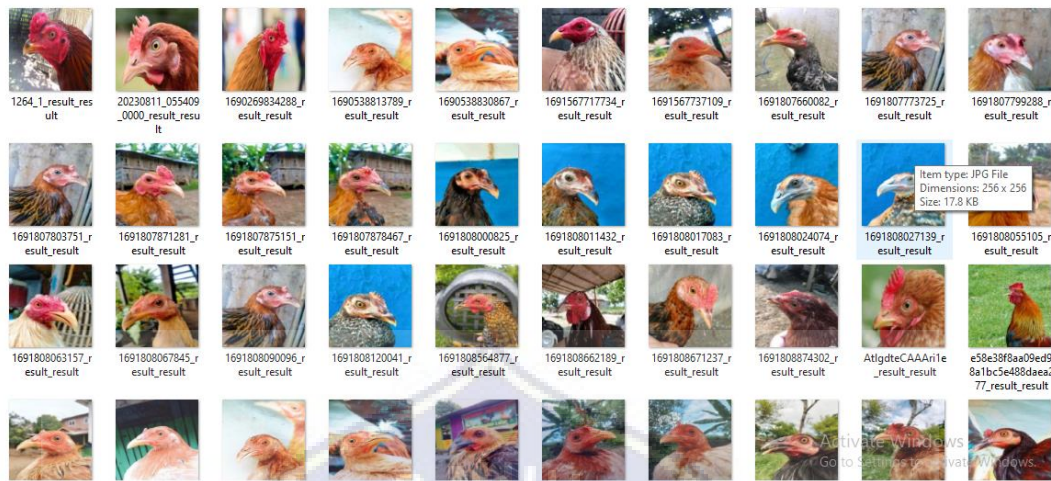


## Kaki Healthy

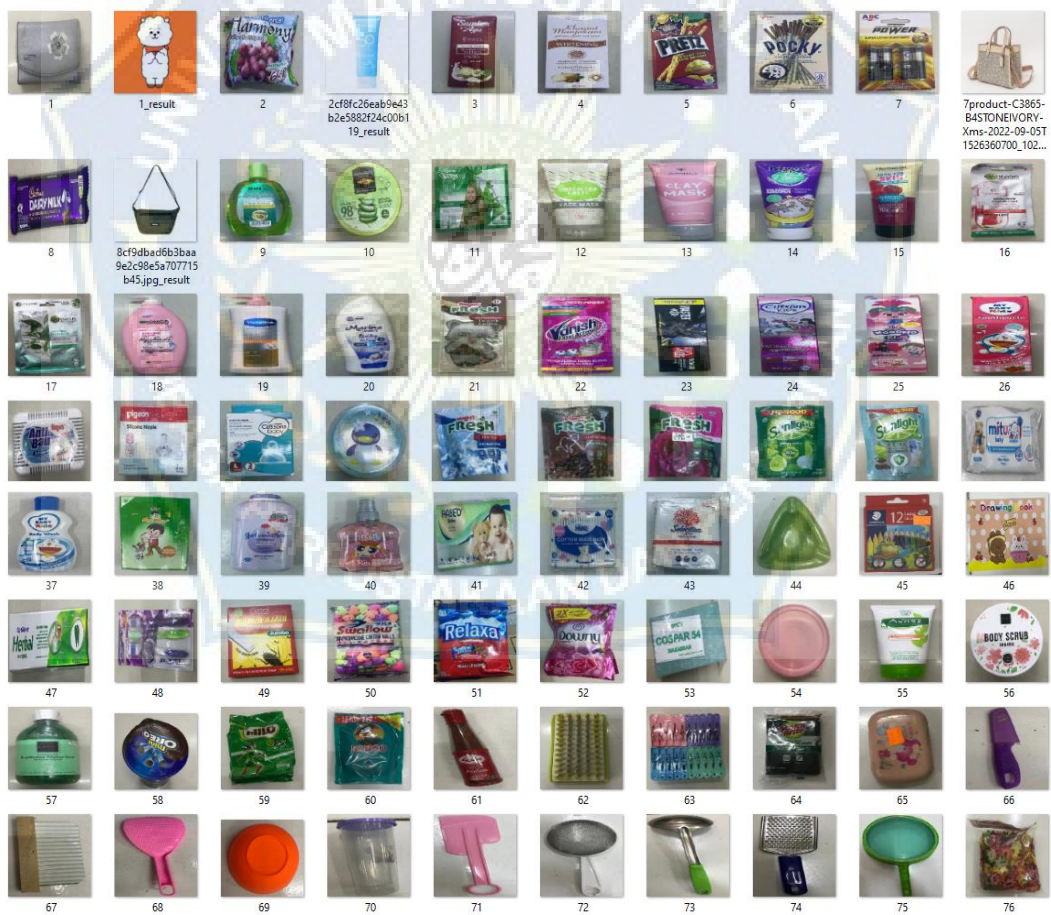




## Mata Healthy

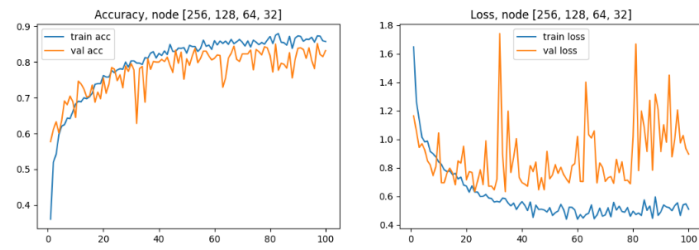


## Data Uji

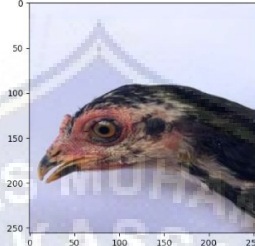


## Lampiran 5. Hasil Pengujian Fungsi Aktivasi relu-softmax dan softmax-relu

### 1. pengujian relu-softmax

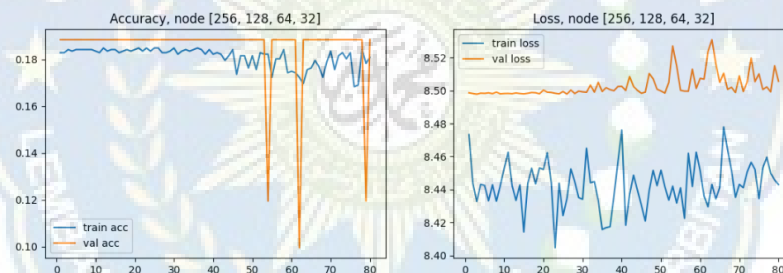


<matplotlib.image.AxesImage at 0x7e02b03e5400>



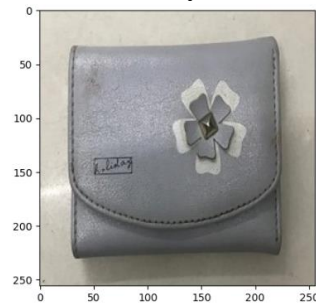
```
[33] y_pred
array([[1.2061044e-11, 1.2240386e-09, 3.3110374e-09, 5.1019965e-03,
        4.8956391e-08, 9.9489802e-01]], dtype=float32)
```

### 2. pengujian softmax-relu



```
[33] y_pred
array([[0.29540297, 0.24114326, 0.40630236, 0.,
        0., 0.26243117,
        0., 0.], dtype=float32)
```

## Lampiran 6. Hasil Pengjian Bukan Data Ayam



```
✓ [32] y_pred  
array([[9.0967793e-01, 8.8572182e-05, 9.2276814e-06, 1.6220512e-05,  
8.0163795e-07, 1.3629732e-17, 7.1859022e-06]], dtype=float32)
```



Lampiran 7. Hasil Scan Plagiasi Per Bab

 **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 : Isnawati  
Nim : 105841110319  
Program Studi : Teknik Informatika  
Dengan nilai:

No	Bab	Nilai	Ambang Batas
1	Bab 1	9 %	10 %
2	Bab 2	24 %	25 %
3	Bab 3	9 %	10 %
4	Bab 4	8 %	10 %
5	Bab 5	4 %	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, 28 Agustus 2023  
Mengetahui  
Kepala UPT Perpustakaan dan Penerbitan,  
  
Isnawati, S.Hum., M.I.P.  
NEM. 964 591

Jl. Sultan Alauddin no 259 makassar 90222  
Telepon (0411)866972,881 593,fax (0411)865 588  
Website: [www.library.unismuh.ac.id](http://www.library.unismuh.ac.id)  
E-mail: [perpustakaan@unismuh.ac.id](mailto:perpustakaan@unismuh.ac.id)



# ●BAB I Isnawati - 105841110319

by Tahap Tutup



**Submission date:** 27-Aug-2023 01:22PM (UTC+0700)

**Submission ID:** 2151893649

**File name:** BAB\_I\_67.docx (15.52K)

**Word count:** 533

**Character count:** 3359

BAB I Isnawati - 105841110319

ORIGINALITY REPORT

9%

SIMILARITY INDEX

9%

INTERNET SOURCES

0%

PUBLICATIONS

0%

STUDENT PAPERS

PRIMARY SOURCES

1

mafiadoc.com

Internet Source

2%

2

repository.ump.ac.id

Internet Source

2%

3

text-id.123dok.com

Internet Source

2%

4

eprints.umm.ac.id

Internet Source

2%

5

es.scribd.com

Internet Source

2%

Exclude quotes On

Exclude bibliography On

Exclude matches < 2%

# BAB II Isnawati - 105841110319

by Tahap Tutup



**Submission date:** 27-Aug-2023 10:45AM (UTC+0700)

**Submission ID:** 2151849193

**File name:** BAB\_II\_72.docx (224.57K)

**Word count:** 2597

**Character count:** 16736



## BAB II Isnawati - 105841110319

### ORIGINALITY REPORT

**24%**

SIMILARITY INDEX

**24%**

INTERNET SOURCES

**6%**

PUBLICATIONS

**15%**

STUDENT PAPERS

### PRIMARY SOURCES

1

lib.unnes.ac.id

Internet Source

5%

2

eprints.unisa.ac.id

Internet Source

2%

3

123dok.com

Internet Source

2%

4

www.scribd.com

Internet Source

2%

5

repository.amikom.ac.id

Internet Source

2%

6

Submitted to Universitas Riau

Student Paper

2%

7

Submitted to Clarkston Community Schools

Student Paper

2%

8

repository.unja.ac.id

Internet Source

2%

9

sipora.polije.ac.id

Internet Source

2%

10

tifaniaprilia.blogspot.com

Internet Source

2%

11

budidayaternak.id

Internet Source

2%

12

ejournal.unsrat.ac.id

Internet Source

2%

Exclude quotes

On

Exclude matches

< 2%

Exclude bibliography

On



# BAB III Isnawati - 105841110319

by Tahap Tutup



**Submission date:** 27-Aug-2023 01:23PM (UTC+0700)

**Submission ID:** 2151893835

**File name:** BAB\_III\_79.docx (45.2K)

**Word count:** 381

**Character count:** 2339



BAB III Isnawati - 105841110319

ORIGINALITY REPORT

9%

SIMILARITY INDEX

7%

INTERNET SOURCES

9%

PUBLICATIONS

4%

STUDENT PAPERS

PRIMARY SOURCES

1

Syaiful Bahri. "Respon Masyarakat Terhadap Madrasah Takmiliah Di Kabupaten Rejang Lebong", Tadris : Jurnal Studi Manajemen Pendidikan, 2018

Publication

4%

2

core.ac.uk  
Internet Source

3%

3

Suri Amilia, Syardiansah Syardiansah. "Analisis Bauran Pemasaran Kerajinan Anyaman Tikar Pandan Terhadap Kepuasan Konsumen", Jurnal Manajemen Motivasi, 2021

Publication

2%

Exclude quotes On

Exclude matches < 2%

Exclude bibliography On

# BAB IV Isnawati - 105841110319

by Tahap Tutup



**Submission date:** 27-Aug-2023 01:23PM (UTC+0700)

**Submission ID:** 2151894015

**File name:** BAB\_IV\_74.docx (979.73K)

**Word count:** 2105

**Character count:** 14194

## BAB IV Isnawati - 105841110319

### ORIGINALITY REPORT

8%

SIMILARITY INDEX

8%

INTERNET SOURCES

2%

PUBLICATIONS

0%

STUDENT PAPERS

### PRIMARY SOURCES

1

repository.unja.ac.id  
Internet Source

8%



Exclude quotes On

Exclude bibliography On

Exclude matches 2%





# BAB V Isnawati - 105841110319

by Tahap Tutup



**Submission date:** 27-Aug-2023 01:24PM (UTC+0700)

**Submission ID:** 2151894165

**File name:** BAB\_V\_72.docx (15.07K)

**Word count:** 177

**Character count:** 1131



## BAB V Isnawati - 105841110319

### ORIGINALITY REPORT

4%

SIMILARITY INDEX

4%

INTERNET SOURCES

0%

PUBLICATIONS

0%

STUDENT PAPERS

### PRIMARY SOURCES

1

pt.scribd.com

Internet Source

4%



Exclude quotes On

Exclude bibliography On

Exclude matches < 2%

