

**KLASIFIKASI TELUR FERTIL DAN INFERTIL MENGGUNAKAN  
SEGMENTASI GAMBAR DENGAN ALGORITMA (CNN)**

**SKRIPSI**

**DISUSUN OLEH**

**JUAN SAMUDRA**

**NPM.200901016**



**PROGRAM STUDI SISTEM INFORMASI  
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI  
UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA**

**MEDAN**

**2024**

**KLASIFIKASI TELUR FERTIL DAN INFERTIL MENGGUNAKAN  
SEGMENTASI GAMBAR DENGAN ALGORITMA (CNN)**

**SKRIPSI**

**Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer  
(S.Kom) dalam Program Studi Sistem Informasi pada Fakultas Ilmu Komputer  
dan Teknologi Informasi, Universitas Muhammadiyah Sumatera Utara**

**JUAN SAMUDRA**

**NPM.200901016**

**PROGRAM STUDI SISTEM INFORMASI  
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI  
UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA**

**MEDAN**

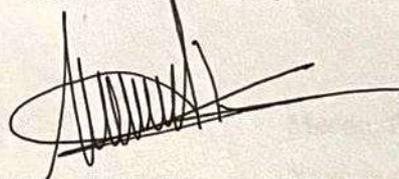
**2024**

## LEMBAR PENGESAHAN

Judul Skripsi : KLASIFIKASI TELUR FERTIL DAN INFERTIL  
MENGUNAKAN SEGMENTASI GAMBAR  
DENGAN ALGORITMA (CNN)

Nama Mahasiswa : JUAN SAMUDRA  
NPM : 2009010116  
Program Studi : SISTEM INFORMASI

Menyetujui  
Komisi Pembimbing



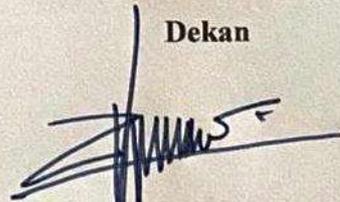
(Martiano S.Pd, S.Kom., M.Kom)  
NIDN. 0128029302

Ketua Program Studi



(Martiano S.Pd, S.Kom., M.Kom)  
NIDN. 0128029302

Dekan



(Dr. Al-Khowarizmi, S.Kom., M.Kom.)  
NIDN. 0127099201

**PERNYATAAN ORISINALITAS**

**KLASIFIKASI TELUR FERTIL DAN INFERTIL MENGGUNAKAN  
SEGMENTASI GAMBAR DENGAN ALGORITMA (CNN)**

**SKRIPSI**

Saya menyatakan bahwa karya tulis ini adalah hasil karya sendiri, kecuali beberapa kutipan dan ringkasan yang masing-masing disebutkan sumbernya.

Medan, Bulan 2024

Yang membuat pernyataan



Juan Samudra

NPM. 2009010116

**PERNYATAAN PERSETUJUAN PUBLIKASI**

**KARYA ILMIAH UNTUK KEPENTINGAN  
AKADEMIS**

Sebagai sivitas akademika Universitas Muhammadiyah Sumatera Utara, saya bertanda tangan dibawah ini:

Nama : Juan Samudra  
NPM : 2009010116  
Program Studi : Sistem Informasi  
Karya Ilmiah : Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Muhammadiyah Sumatera Utara Hak Bedas Royalti Non-Eksekutif (*Non-Exclusive Royalty free Right*) atas penelitian skripsi saya yang berjudul:

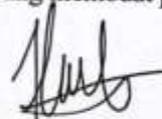
**KLASIFIKASI TELUR FERTIL DAN INFERTIL MENGGUNAKAN  
SEGMENTASI GAMBAR DENGAN ALGORITMA (CNN)**

Beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Non-Eksekutif ini, Universitas Muhammadiyah Sumatera Utara berhak menyimpan, mengalih media, memformat, mengelola dalam bentuk database, merawat dan mempublikasikan Skripsi saya ini tanpa meminta izin dari saya selama tetap mencantumkan nama saya sebagai penulis dan sebagai pemegang dan atau sebagai pemilik hak cipta.

Demikian pernyataan ini dibuat dengan sebenarnya.

Medan, Bulan 2024

Yang membuat pernyataan



Juan Samudra

NPM.2009010116

## RIWAYAT HIDUP

### DATA PRIBADI

Nama Lengkap : Juan Samudra  
Tempat dan Tanggal Lahir : Binjai, 03 Mei 2002  
Alamat Rumah : JL.P.KEMERDEKAAN LK V NO 07  
Telepon/Faks/HP : 089522804801  
E-mail : juansamudralbss@gmail.com  
Instansi Tempat Kerja : Belum Bekerja  
Alamat Kantor : -

### DATA PENDIDIKAN

SD : Gajah Mada Binjai TAMAT: 2014  
SMP : SMP Negeri 10 Binjai TAMAT: 2017  
SMA : SMA Negeri 1 Binjai TAMAT: 2020

## KATA PENGANTAR



Puji syukur Alhamdulillah, Penulis panjatkan kehadiran Allah SWT., yang telah melimpahkan banyak rahmat dan karunia-Nya serta memberi kekuatan kepada Penulis untuk yang menuntaskan tugas akhir dalam meraih Strata 1 ini. Skripsi ini Penulis sajikan dalam bentuk buku yang sederhana. Judul Skripsi pada penelitian ini adalah sebagai berikut. **“KLASIFIKASI TELUR FERTIL DAN INFERTIL MENGGUNAKAN SEGMENTASI GAMBAR DENGAN ALGORITMA (CNN)”**.

Adapun Tujuan penulisan skripsi ini dibuat sebagai salah satu syarat kelulusan Program Strata Satu (S1) Sistem Informasi Universitas Muhammadiyah Sumatera Utara. Sebagai bahan penulisan diambil berdasarkan hasil penelitian, observasi dan beberapa sumber literatur yang mendukung penulisan ini. Penulis menyadari bahwa tanpa bimbingan dan dorongan dari semua pihak, maka Penulis skripsi ini tidak akan lancar oleh kerana itu pada kesempatan ini, izinkanlah Penulis menyampaikan ucapan terimakasih kepada :

1. Bapak Prof. Dr. Agussani, M.AP., Rektor Universitas Muhammadiyah Sumatera Utara (UMSU).
2. Bapak Dr. Al-Khowarizmi, S.Kom., M.Kom sebagai Dekan Fakultas Ilmu Komputer dan Teknologi Informasi.
3. Bapak Martiano, S.Kom., M.Kom sebagai Ketua Prodi Sistem Informasi yang selalu memberikan dukungan.

4. Bapak Martiano, S.Kom., M.Kom selaku Dosen Pembimbing saya yang selalu memberikan pengingat untuk selalu konsisten mengerjakan skripsi. viii
5. Teristimewa kepada Ayahanda Hendra Vota dan Ibunda Zuniar sebagai Orang Tua Penulis yang tak pernah lelah mendoakan, mengusahakan dan memberikan fasilitas serta dukungan terbaik secara material maupun non-material untuk kesuksesan penulis.
6. Tersayang kepada Chintya Sari Kartika Dewi yang merupakan seseorang yang selalu mensupport dibalik layar dan selalu menjadi motivasi penulis menjadi rajin, tekun untuk menyelesaikan pendidikan.
7. Seluruh Staff dan Keanggotaan Biro Kemahasiswaan yang mendukung dalam proses pengerjaan Penelitian ini,
8. Untuk teman seperjuangan beserta seluruh kelas C1sisfo, terimakasih atas segala bantuan, waktu, support dan kebaikan yang diberikan kepada penulis disaat masa sulit mengerjakan skripsi ini.
9. Seluruh teman-teman Angkatan Sistem Informasi 2020 yang telah sama-sama berjuang.
10. Kepada Diri sendiri terimakasih atas kerjasamanya yang begitu luar biasa berjuang dalam mengupayakan sebaik dan sebisa mungkin untuk bisa melewati suka dan duka secara penuh penerimaan pada saat proses pengerjaan skripsi. Serta semua pihak terlibat yang namanya tidak bisa disebutkan satu persatu sehingga terwujudnya penulisan ini. Penulis menyadari bahwa penulisan Skripsi ini jauh dari kata sempurna, untuk itu Penulis memohon kritik dan saran yang bersifat membangun demi kesempurnaan penulisan ini di mendatang.

# KLASIFIKASI TELUR FERTIL DAN INFERTIL MENGGUNAKAN SEGMENTASI GAMBAR DENGAN ALGORITMA (CNN)

## ABSTRAK

Telur juga merupakan salah satu sumber pangan yang optimal karena juga mengandung beragam nutrisi yang dibutuhkan oleh makhluk hidup yang sangat bermanfaat seperti protein, lemak, vitamin, dan mineral tidak kalah dari kandungan daging ayam. Sehingga telur sangat sering kali digunakan dalam berbagai olahan pangan atau juga digunakan sebagai lauk pangan untuk di konsumsi oleh makhluk hidup. Perlu diketahui tidak semua telur ayam dapat dikonsumsi oleh masyarakat dikarenakan telur memiliki 2 jenis, yaitu telur fertil dan telur infertil. Telur fertil adalah telur yang memiliki potensi untuk menetas karena mengalami perkembangan embrio, yang ditandai dengan kehadiran bintik darah atau embrio. Pada telur jenis ini biasanya tidak di jual atau tidak di konsumsi pada masyarakat dikarenakan sebagai bibit pengembangbiakan anak ayam baru. Sementara itu, telur ayam infertil adalah jenis telur yang tidak akan menetas karena tidak mengalami perkembangan embrio selama proses penetasan.

Kemajuan teknologi telah memberikan dampak yang signifikan pada berbagai sektor, termasuk dalam peternakan ayam. Saat ini teknologi telah menghasilkan perkembangan yang pesat dalam bidang tersebut, memungkinkan peternak untuk meningkatkan efisiensi dari segi waktu dan tenaga untuk mengklasifikasi telur salah satunya dengan menggunakan segmentasi citra atau gambar. Klasifikasi telur ayam yang tidak subur penting dilakukan terutama oleh peternak untuk memilih telur fertil dan telur infertil. Saat ini, metode pemeriksaan telur ayam fertil dan infertil masih menggunakan cara yang kurang efektif, yaitu hanya dengan meneropong cangkang telur, yang disebut *candling*. Dalam proses ini tingkat kegagalan peternak dalam membedakan telur fertil dan infertil terbilang rendah yaitu berkisar 2%-5% saja, namun metode ini tidak selalu akurat karena tidak semua peternak dapat melihat dengan jelas hasil *candling*, yang dapat mengakibatkan kesalahan dalam prediksi. Dengan menerapkan Algoritma *Convolutional Neural Network* (CNN).

Menggunakan algoritma CNN dengan model VGG16 yang dilatih pada gambar grayscale telur untuk mengklasifikasikan telur fertil dan infertil. menunjukkan akurasi 100% dalam klasifikasi telur fertil dan infertil. Hal ini tentunya menunjukkan kemampuan model dalam membedakan antara telur fertil dan infertil dengan sangat baik. Dengan merancang algoritma CNN melibatkan penggunaan arsitektur VGG16 untuk mengekstraksi fitur dari gambar grayscale telur. Dataset gambar telur fertil dan infertil digunakan untuk menguji model di Google Colab. Dengan menggunakan *matrix confusion*, evaluasi memastikan bahwa model dapat mengklasifikasikan telur dengan sangat akurat.

Kata Kunci: Telur ayam; Convolutional Neural Network (CNN), Machine learning.

# CLASSIFICATION OF FERTILE AND INFERTILE EGGS USING IMAGE SEGMENTATION WITH (CNN) ALGORITHM

## ABSTRACT

Eggs are also one of the optimal food sources because they also contain a variety of nutrients needed by living things that are very useful such as protein, fat, vitamins, and minerals no less than the content of chicken meat. So that eggs are very often used in various food preparations or also used as food side dishes for consumption by living things. Please note that not all chicken eggs can be consumed by the community because eggs have 2 types, namely fertile eggs and infertile eggs. Fertile eggs are eggs that have the potential to hatch because they experience embryonic development, which is characterized by the presence of blood spots or embryos. This type of egg is usually not sold or not consumed in the community because it is a seed for breeding new chicks. Meanwhile, infertile chicken eggs are a type of egg that will not hatch because it does not experience embryonic development during the hatching process.

Technological advances have had a significant impact on various sectors, including chicken farming. Currently, technology has produced rapid developments in the field, enabling farmers to increase efficiency in terms of time and energy to classify eggs, one of which is by using image segmentation. Classification of infertile chicken eggs is important especially by farmers to select fertile eggs and infertile eggs. Currently, the method of examining fertile and infertile chicken eggs still uses a less effective way, namely only by looking at the eggshell, which is called candling. In this process, the failure rate of farmers in distinguishing fertile and infertile eggs is fairly low, which is around 2%-5%, but this method is not always accurate because not all farmers can clearly see the results of candling, which can lead to errors in prediction. By applying the Convolutional Neural Network (CNN) algorithm.

Using the CNN algorithm with the VGG16 model trained on grayscale images of eggs to classify fertile and infertile eggs. showed 100% accuracy in the classification of fertile and infertile eggs. This certainly shows the ability of the model to distinguish between fertile and infertile eggs very well. Designing the CNN algorithm involved using the VGG16 architecture to extract features from grayscale images of eggs. A dataset of fertile and infertile egg images was used to test the model on Google Colab. Using confusion matrix, the evaluation confirmed that the model can classify eggs very accurately.

Keywords: Chicken eggs; Convolutional Neural Network (CNN),  
Machinelearning.

## DAFTAR ISI

LEMBAR PENGESAHAN .....	ii
PERNYATAAN ORISINALITAS.....	Error! Bookmark not defined.
PERNYATAAN PERSETUJUAN PUBLIKASI.....	iv
RIWAYAT HIDUP .....	v
KATA PENGANTAR.....	vi
ABSTRAK .....	viii
ABSTRACT .....	ix
DAFTAR ISI.....	x
DAFTAR TABEL .....	xiii
DAFTAR GAMBAR.....	xiv
BAB I.....	1
PENDAHULUAN .....	1
1.1. Latar Belakang Masalah .....	1
1.2. Rumusan Masalah .....	5
1.3. Batasan Masalah .....	5
1.4. Tujuan Penelitian.....	5
1.5. Manfaat Penelitian.....	5
BAB II.....	7
LANDASAN TEORI .....	7
2.1 Penelitian Terkait.....	7
2.2. Landasan Teori .....	8
2.2.1 Klasifikasi .....	8
2.2.2 Telur .....	9
2.3 Kecerdasan Buatan (Artificial Intelligence/AI).....	11
2.4 Pengolahan Citra (Image Processing) .....	14
2.5 Segmentasi Gambar (Image Segmentation) .....	16
2.6 Jaringan Syaraf Tiruan ( <i>Artificial Neural Network</i> ) .....	16
2.7 Convolutional Neural Network (CNN).....	18
2.8 Tensorflow .....	23
2.9 Keras .....	24
2.10 Python .....	24
2.11 Google Colaboratory.....	25
BAB III.....	26
METODOLOGI PENELITIAN .....	26

3.1	Gambaran Umum Objek Penelitian .....	26
3.2	UML (Unified Modelling Language).....	27
3.2.1	Use Case Diagram .....	27
3.3	Konsep Pemodelan Sistem.....	28
3.3.1	Flowchart .....	28
3.4	Data Collection .....	29
3.5	Preprocessing Data .....	30
3.6	Grayscale .....	31
3.7	Image Enchanment .....	32
3.8	Pemodelan Klasifikasi Menggunakan CNN .....	33
3.9	Data Training .....	36
3.10	Data Testing.....	37
3.11	Evaluation Model .....	37
BAB IV .....		39
HASIL DAN PEMBAHASAN.....		39
4.1.	Hasil Pengumpulan Data Set .....	39
4.2.	Implementasi Model CNN.....	41
4.2.1.	Persiapan Dataset.....	41
4.2.2.	Menghubungkan Google Drive.....	44
4.2.3.	Import Library.....	44
4.2.4.	Import Arsitektur VGG16 .....	45
4.2.5.	Pemeriksaan Direktori dan File Dataset.....	46
4.2.6.	Visualisasi Dataset.....	47
4.2.7.	Dataframe Dataset .....	47
4.2.8.	Split Data .....	48
4.2.9.	Data Generator.....	48
4.2.10.	Visualisasi Data Generator .....	50
4.2.11.	Model VGG16 Greyscale.....	51
4.2.12.	Konfigurasi Model Pelatihan .....	52
4.2.13.	Training Model.....	52
4.2.14.	Evaluasi Model.....	53
4.2.15.	Grafik Accuracy dan Loss.....	55
4.2.16.	Testing Model.....	56
4.2.17.	Save Model.....	59
4.3.	Implementasi Website.....	59
4.3.1.	Tampilan Halaman Utama (Index) .....	60
4.3.2.	Tampilan Upload Sukses .....	60
4.3.3.	Tampilan Upload Gagal .....	61

4.3.4.	Tampilan Halaman Hasil (Sukses) .....	62
4.4.	Pengujian Blackbox Testing.....	62
4.4.1.	Pengujian Upload JPEG.....	63
4.4.2.	Pengujian Upload JPG .....	64
4.4.3.	Pengujian Upload JFIF .....	64
4.4.4.	Pengujian Upload PNG .....	65
4.4.5.	Pengujian Upload DOC.....	66
4.4.6.	Pengujian Upload PDF .....	66
4.4.7.	Pengujian Upload EXCEL .....	67
4.4.8.	Pengujian Upload MP4.....	67
4.4.9.	Pengujian Klasifikasi Fertl.....	68
4.4.10.	Pengujian Klasifikasi Infertil.....	69
BAB V	.....	72
PENUTUP	.....	72
5.1	Kesimpulan.....	72
5.2	Saran .....	72
DAFTAR PUSTAKA	.....	73
LAMPIRAN	.....	76

## DAFTAR TABEL

	<b>HALAMAN</b>
TABEL 2.1. Tabel Penelitian Terdahulu	7
TABEL 3.1. Tabel Data Set	37
TABEL 3.2. Tabel Data Set Training	37
TABEL 4.1. Tabel Data Set	39
TABEL 4.2. Tabel Pengujian Blackbox Testing	69

## DAFTAR GAMBAR

	<b>HALAMAN</b>	
GAMBAR 2.1.	Ciri Telur fertil	10
GAMBAR 2.2.	Citra Telur Fertil	11
GAMBAR 2.3.	Bagian Utama Artificial Intelligence	13
GAMBAR 2.4.	Cara Kerja Segmentasi Gambar	16
GAMBAR 2.5.	Struktur Dasar Jaringan Syaraf Tiruan	17
GAMBAR 2.6.	Arsitektur CNN	19
GAMBAR 2.7.	Convolutional Layer	20
GAMBAR 2.8.	Proses Convolutional Layer	20
GAMBAR 2.9.	Ilustrasi Pooling Layer	22
GAMBAR 2.10.	Fully Connected Layer	23
GAMBAR 2.11.	Tensorflow	23
GAMBAR 2.12.	Python	25
GAMBAR 3.1.	Teknik Perancangan Sistem	27
GAMBAR 3.2.	Use Case Diagram	28
GAMBAR 3.3.	Flowchart Sistem	29
GAMBAR 3.4.	Teknik Pengambilan Data Set Telur	31
GAMBAR 3.5.	Proses Grayscale Pada Telur	32
GAMBAR 3.6.	Perbandingan citra Grayscale dengan Clahe dan He	33
GAMBAR 3.7.	Arsitektur CNN	33
GAMBAR 4.1.	Dataset Telur Fertil	40
GAMBAR 4.2.	Dataset Telur Infertil	41
GAMBAR 4.3.	Folder Kelas	42
GAMBAR 4.4.	Infertil Pada Google Drive	43
GAMBAR 4.5.	Fertil Pada Google Drive	43
GAMBAR 4.6.	Menghubungkan Google Drive	44
GAMBAR 4.7.	Import Library	45
GAMBAR 4.8.	Import VGG16	46
GAMBAR 4.9.	Pemeriksaan Dataset	46
GAMBAR 4.10.	Visualisasi Dataset	47
GAMBAR 4.11.	Dataframe Dataset	47
GAMBAR 4.12.	Split Data	48
GAMBAR 4.13.	Data Generator	49
GAMBAR 4.14.	Visualisasi Data Generator	50
GAMBAR 4.15.	Sebelum Greyscale	50
GAMBAR 4.16.	Sesudah Greyscale	51
GAMBAR 4.17.	Model VGG16 Greyscale	51
GAMBAR 4.18.	Konfigurasi Model	52
GAMBAR 4.19.	Training Epoch	53
GAMBAR 4.20.	Evaluasi Model	54
GAMBAR 4.21.	Analisis Hasil Evaluasi Model dan Visualisasi	55
GAMBAR 4.22.	Pengecekan Akurasi Model pada Data Pengujian	56
GAMBAR 4.23.	Grafik Akurasi dan loss	56
GAMBAR 4.24.	Confusion Matriks	57

GAMBAR 4.25.	Report Classification	58
GAMBAR 4.26.	Save Model	59
GAMBAR 4.27.	Halaman Utama	60
GAMBAR 4.28.	Upload Sukses	60
GAMBAR 4.29.	Upload Gagal	61
GAMBAR 4.30.	Hasil Klasifikasi	62
GAMBAR 4.31.	Pengujian JPEG	63
GAMBAR 4.32.	Pengujian Jpg	64
GAMBAR 4.33.	Pengujian Jfif	64
GAMBAR 4.34.	Pengujian Png	65
GAMBAR 4.35.	Pengujian Doc	66
GAMBAR 4.36.	Pengujian Pdf	66
GAMBAR 4.37.	Pengujian Excel	67
GAMBAR 4.38.	Pengujian Mp4	67
GAMBAR 4.39.	Pengujian Klasifikasi Fertile	68
GAMBAR 4.40.	Pengujian Klasifikasi Infertile	69

# **BAB I**

## **PENDAHULUAN**

### **1.1. Latar Belakang Masalah**

Ayam adalah salah satu sumber utama protein bagi manusia. Meskipun sekarang ada banyak sumber protein lain selain daging ayam, masyarakat cenderung lebih memilih daging ayam sebagai sumber protein utama. Dengan pertumbuhan populasi manusia yang terus meningkat, kebutuhan akan daging ayam juga semakin tinggi. Tingginya permintaan akan daging ayam itu akan berhambas pada kenaikan harga daging ayam itu sendiri, yang membuat masyarakat mencari alternatif lain selain daging ayam, yang terjangkau dalam harga dan mudah ditemukan untuk di konsumsi selalu dalam keseharian masyarakat (Ramdan Ahaya, 2018). Telur juga merupakan salah satu sumber pangan yang optimal karena juga mengandung beragam nutrisi yang dibutuhkan oleh makhluk hidup yang sangat bermanfaat seperti protein, lemak, vitamin, dan mineral tidak kalah dari kandungan daging ayam. Kandungan protein berkualitas tinggi tersebut yang terdapat didalam telur disebabkan oleh keberadaan susunan asam amino esensial yang lengkap, sehingga telur sangat sering kali digunakan dalam berbagai olahan pangan atau juga digunakan sebagai lauk pangan untuk di konsumsi oleh makhluk hidup (I Ketut Suada, 2012). Di masyarakat telur dapat sangat mudah diperoleh dan harganya relatif murah atau terjangkau bagi masyarakat. Ketersediaan telur yang mudah diperoleh dan terjangkau di seluruh penjuru wilayah ini memiliki keuntungan sendiri dalam bermasyarakat karena dapat memenuhi kebutuhan pangan masyarakat setiap hari (I Ketut Suada, 2012).

Perlu diketahui tidak semua telur ayam dapat dikonsumsi oleh masyarakat dikarenakan telur memiliki 2 jenis, yaitu telur fertil dan telur infertil. Telur fertil adalah telur yang memiliki potensi untuk menetas karena mengalami perkembangan embrio, yang ditandai dengan kehadiran bintik darah atau embrio. Pada telur jenis ini biasanya tidak di jual atau tidak di konsumsi pada masyarakat dikarenakan sebagai bibit pengembangbiakan anak ayam baru. Sementara itu, telur ayam infertil adalah jenis telur yang tidak akan menetas karena tidak mengalami perkembangan embrio selama proses penetasan, pada telur jenis ini biasa yang digunakan masyarakat untuk di konsumsi dan di perjual belikan. (Firdaus, 2021). Penetasan telur dilakukan dengan tujuan untuk menghasilkan telur yang mengandung embrio yang berkembang, ditandai dengan adanya *bloodspot* (noktah darah) dan *blood vessel* (pembuluh darah), yang dikenal sebagai telur fertil. Sebaliknya, telur yang tidak mengandung embrio disebut sebagai telur infertil. Proses penetasan telur bertujuan untuk menghasilkan telur fertil dengan embrio yang berkembang, yang secara khusus ditandai oleh kehadiran *bloodspot* dan pembuluh darah, sementara telur infertil tidak mengandung embrio (Firdaus, 2021).

Permasalahan utama pada penelitian ini adalah sulit membedakan citra candling telur ayam fertil dan infertil. Kesalahan identifikasi pada citra candling dikarenakan hasil peneropongan atau pengamatan yang kurang jelas atau ciri pada telur terlihat samar sehingga menafsirkan salah atau memang telur tersebut subur tetapi sel embrio mati pada proses pengeraman. Penyebab embrio mati adalah keracunan bakteri dan penyebab lain seperti suhu yang tidak teratur pada mesin inkubasi. Sedangkan telur yang benar-benar infertil tidak akan menetas dan kondisinya tidak terjadi pembusukan di dalamnya, begitu

sebaliknya jika telur tersebut fertil dan embrio mati sebelum menetas maka akan terjadi pembusukan bahkan telur meletus. Sehingga dalam kejadian ini sangat disayangkan dan dirugikan apabila telur yang kemungkinan fertil atau telur bakal ditetaskan namun digolongkan ke jenis infertil atau dijadikan telur konsumsi (Mutaqin Akbar,2021).

Kemajuan teknologi telah memberikan dampak yang signifikan pada berbagai sektor, termasuk dalam peternakan ayam. Saat ini teknologi telah menghasilkan perkembangan yang pesat dalam bidang tersebut, memungkinkan peternak untuk meningkatkan efisiensi dari segi waktu dan tenaga untuk mengklasifikasi telur salah satunya dengan menggunakan segmentasi citra atau gambar. Segmentasi adalah salah satu tahapan yang digunakan untuk mengklasifikasi objek dalam sebuah gambar. Proses segmentasi ini memiliki kemampuan untuk membagi citra ke dalam beberapa bagian atau wilayah. Tahapan segmentasi dilakukan setelah tahapan preprocessing. Preprocessing dilakukan untuk meningkatkan kualitas gambar sehingga proses klasifikasi dapat dilakukan dengan lebih baik (Shoffan Saifullah, 2020). Klasifikasi telur ayam yang tidak subur penting dilakukan terutama oleh peternak untuk memilih telur fertil dan telur infertil. Saat ini, metode pemeriksaan telur ayam fertil dan infertil masih menggunakan cara yang kurang efektif, yaitu hanya dengan menepong cangkang telur, yang disebut candling. Dalam proses ini tingkat kegagalan peternak dalam membedakan telur fertil dan infertil terbilang rendah yaitu berkisar 2%-5% saja, namun metode ini tidak selalu akurat karena tidak semua peternak dapat melihat dengan jelas hasil candling, yang dapat mengakibatkan kesalahan dalam prediksi. Dengan menerapkan Algoritma *Convolutional Neural Network* (CNN), peternak

dapat melakukan pemeriksaan dengan lebih efektif. Pemeriksaan dilakukan dengan cara meneropong cangkang telur dan mengolah hasilnya menggunakan CNN dengan menggunakan beberapa dataset yang sudah dikumpulkan melalui studi lapangan. Metode ini mampu mengklasifikasikan telur ayam fertil dan telur ayam infertil (Firdaus, 2021).

Penelitian ini menawarkan solusi terhadap masalah tersebut dengan memanfaatkan teknologi *Computer Vision* yang mampu mengklasifikasi objek berdasarkan karakteristiknya sehingga dapat diklasifikasikan baik dan tepat dengan menggunakan data set yang ada. Kontribusi utama dari penelitian ini adalah pengembangan teknologi yang memfasilitasi pengujian fertilitas telur dengan menggunakan metode *Convolutional Neural Network* (CNN). Metode ini adalah jenis arsitektur jaringan syaraf yang terinspirasi oleh cara kerja visual otak hewan, arsitektur *Convolutional Neural Network* (CNN) terdiri dari lapisan lapisan konvolusi yang dapat mengekstrak fitur-fitur penting dari gambar secara hierarkis, kemudian dilakukan klasifikasi terhadap data gambar telur fertil dan infertil. Teknologi ini diharapkan dapat memberikan kemudahan dan akurasi dalam proses pengujian fertilitas telur ayam (Firdaus, 2021).

## **1.2. Rumusan Masalah**

Rumusan masalah dalam penelitian ini yaitu:

- a. Bagaimana proses algoritma *Convolutional Neural Network* (CNN) untuk mengklasifikasi antara telur ayam fertil dan infertil?
- b. Bagaimana merancang algoritma *Convolutional Neural Network* (CNN) dalam mengklasifikasi telur fertil dan infertil?

## **1.3. Batasan Masalah**

Terdapat beberapa batasan permasalahan dalam penelitian ini yaitu:

- a. Penelitian ini akan difokuskan pada implementasi Algoritma *Convolutional Neural Network* (CNN) dalam klasifikasi telur ayam fertil dan telur ayam infertil
- b. Data set yang akan digunakan adalah telur ayam sebanyak 90-100 butir
- c. Metode *Convolutional Neural Network* (CNN) diterapkan berbasis web

## **1.4. Tujuan Penelitian**

Tujuan dari penelitian ini yaitu:

Tujuan utama dari penelitian ini adalah mengaplikasikan pengolahan citra menggunakan segmentasi gambar dengan metode *Convolutional Neural Network* (CNN) untuk mengklasifikasi telur ayam fertil dan telur ayam infertil

## **1.5. Manfaat Penelitian**

Terdapat beberapa manfaat pada penelitian ini yaitu:

- a. Manfaatnya untuk mengefisienkan waktu dalam mengklasifikasi telur ayam fertil dan infertil

- b. Meningkatkan tingkat akurasi dalam mengklasifikasi telur ayam fertil dan infertil

**BAB II**  
**LANDASAN TEORI**

**2.1 Penelitian Terkait**

Beberapa penelitian terkait penggunaan *Machine Learning* menggunakan metode CNN terkait klasifikasi telur fertil dan infertil.

No.	Referensi	Objek	Metode	Hasil Penelitian
1	Sistem Klasifikasi Telur Ayam Fertil dan Infertil Menggunakan Fitur Tekstur Dan Metode K-Nearest Neighbor Berbasis Raspberry	Klasifikasi Telur fertil dan infertil	KNN	Metode KNN berhasil menunjukkan dalam pengklasifikasian telur fertil dan infertil menggunakan <i>leave one out cross validation</i> memperoleh 93,33% nilai akurasi.
2	Klasifikasi Telur Fertil Dan Infertil Menggunakan Jaringan Saraf Tiruan Multilayer Perceptron Berdasarkan Ekstraksi Fitur Warna Dan Bentuk	Klasifikasi Telur fertil dan infertil	MLP	Metode MLP ( <i>Multilayer Perceptron</i> ) mampu melakukan pengujian pada telur yang mencapai tingkat akurasi 98% untuk data yang telah di latih dan 96% untuk data yang belum pernah dilatih.
3	Penerapan Algoritma Convolutional Neural Network dalam Klasifikasi Telur Ayam Fertil dan Infertil Berdasarkan Hasil Candling	Penerapan CNN untuk klasifikasi telur fertil dan infertil	CNN	Penerapan Metode CNN dalam klasifikasi telur ayam fertil dan infertil berhasil mendapatkan nilai akurasi mencapai 98% dan nilai error hanya sebesar 5%

**Tabel 2.1 Penelitian Terdahulu**

## **2.2. Landasan Teori**

### **2.2.1 Klasifikasi**

Pengolahan data dalam dunia digital saat ini merupakan suatu hal yang penting, berbagai data yang diolah akan menghasilkan informasi yang efektif, efisien dan akurat dalam hal ini untuk memenuhi kebutuhan dalam pengolahan data. Sejalan dengan perkembangan digital saat ini muncul beragam metode pengolahan data yang digunakan sesuai dengan tipe data yang akan diolah. Salah satu metode nya adalah Klasifikasi, Klasifikasi merupakan cara pengelompokkan data sesuai dengan ciri-ciri atau karakteristik data yang telah diolah.

Klasifikasi juga merupakan proses pengelompokan objek ke dalam kategori-kategori berdasarkan karakteristik yang sama. Proses klasifikasi dapat dilakukan dengan beberapa cara, baik secara manual maupun dengan bantuan teknologi. Klasifikasi manual dilakukan oleh manusia tanpa bantuan algoritme komputer yang cerdas. Sebaliknya, klasifikasi dengan bantuan teknologi menggunakan berbagai algoritma, termasuk Naïve Bayes, Support Vector Machine, Decision Tree, Fuzzy Logic, dan Neural Network (Wibawa et al., 2018)

Metode manual biasanya bergantung pada keahlian dan pengalaman manusia untuk mengidentifikasi dan mengklasifikasikan objek berdasarkan karakteristik yang dapat diamati. Proses ini dapat memakan waktu dan rentan terhadap kesalahan manusia. Sebaliknya, klasifikasi berbantuan teknologi menggunakan algoritme yang mampu memproses data dalam jumlah besar dengan kecepatan dan ketepatan yang luar biasa. Algoritma seperti Naïve Bayes dan Support Vector Machine menggunakan prinsip-prinsip statistik untuk membuat

prediksi berdasarkan data yang ada, sedangkan Decision Tree menggunakan seperangkat aturan yang berasal dari data pelatihan untuk membuat keputusan. Logika fuzzy adalah metode klasifikasi yang menangani ketidakpastian dan tingkat keanggotaan, sedangkan jaringan saraf adalah jenis kecerdasan buatan yang meniru cara kerja otak manusia untuk mengenali pola dalam data yang kompleks.

Seiring dengan kemajuan teknologi, metode klasifikasi canggih ini menjadi semakin penting di berbagai bidang, seperti pengenalan pola, pemrosesan bahasa alami, dan analisis data besar. Metode-metode ini membantu meningkatkan efisiensi dan akurasi dalam mengklasifikasikan data, sehingga memungkinkan aplikasi yang lebih luas dan lebih berguna dalam kehidupan sehari-hari.

### **2.2.2 Telur**

Telur ayam merupakan sumber protein hewani yang sangat baik dan juga mengandung banyak nutrisi penting lainnya, telur juga sangat umum dikonsumsi oleh masyarakat banyak sebagai asupan pangan setiap hari. Telur sangat mudah ditemukan dan harganya relatif murah dibandingkan asupan protein yang lain seperti daging ayam, daging sapi ataupun daging ikan, karena faktor itulah telur sangat disukai sebagian banyak masyarakat sebagai salah satu pokok pangan. Telur berdasarkan tingkat fertilitas dapat dibedakan menjadi dua, yaitu :

#### **a) Telur Fertil**

Telur fertil adalah telur yang memiliki potensi untuk menetas karena mengalami perkembangan embrio, yang ditandai dengan kehadiran *bloodspot* (noktah darah) dan *blood vessel* (pembuluh darah). Pada telur jenis ini biasanya tidak di jual atau tidak di konsumsi pada masyarakat dikarenakan telur fertil ini

dapat dilanjutkan untuk tahap inkubasi (pengeraman) di dunia industri peternakan. Ciri telur fertil dapat dilihat pada gambar 2.1.



**Gambar 2.1 Ciri telur fertil**

#### **b) Telur Infertil**

Telur ayam infertil adalah jenis telur yang tidak akan menetas karena tidak mengalami perkembangan embrio selama proses penetasan, itu terjadi karena diakibatkan oleh pengaturan temperatur suhu yang salah dan lain sebagainya. Telur infertil juga dapat menjadi tempat berkembangbiaknya bakteri, ciri-ciri lain telur infertil adalah kuning telur bercampur dengan putih telur, tekstur keseluruhan telurnya akan cenderung cair. (Nawawi et al., 2015). Citra telur infertil dapat dilihat pada gambar 2.2.



**Gambar 2.2 Citra telur infertil**

### **2.3 Kecerdasan Buatan (Artificial Intelligence/AI)**

Kecerdasan Buatan (*Artificial Intelligence/AI*) merupakan sebuah dalam bidang ilmu komputer yang menirukan cara berfikir manusia dengan bantuan teknologi komputer untuk memvisualisasikannya. Kecerdasan buatan mengacu kepada sistem komputer yang dapat menunjukkan pengetahuan dan perilaku manusia, seperti kemampuan untuk belajar, menganalisis, menyelesaikan masalah, mengingat, dan memahami bahasa alami manusia (Chandra Halim, 2018).

Dalam beberapa buku dan penulis juga menulis tentang AI bahwa istilah “kecerdasan buatan” didefinisikan sebagai sistem yang menggabungkan perangkat keras dan perangkat lunak yang canggih dengan database yang rumit dan model pemrosesan berbasis pengetahuan untuk mendemonstrasikan karakteristik pengambilan keputusan manusia yang efektif. Pengambilan keputusan yang efektif. Kriteria untuk sistem buatan meliputi berikut ini:

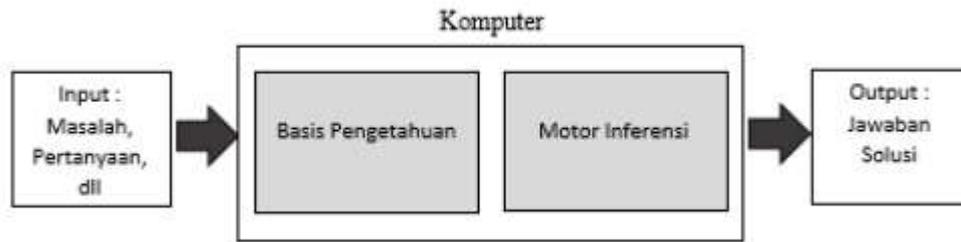
1. Fungsional: sistem harus mampu melakukan fungsi yang telah dirancang untuknya
2. Mampu diproduksi: sistem harus mampu diproduksi oleh manufaktur yang ada proses

3. Dapat dirancang: desain sistem harus dapat dapat dibayangkan oleh desainer yang bekerja dalam konteks budaya mereka konteks budaya mereka
4. Dapat dipasarkan: sistem harus dianggap dapat melayani beberapa tujuan dengan cukup baik, jika dibandingkan dengan pendekatan yang bersaing, untuk menjamin desain dan pembuatannya.

Kecerdasan buatan secara dramatis mengurangi atau menghilangkan risiko bagi manusia dalam banyak aplikasi. Kecerdasan buatan yang kuat perangkat lunak intelijen membantu untuk sepenuhnya mengembangkan kemampuan mesin presisi tinggi robot, sering kali membebaskan mereka dari kontrol manusia langsung dan sangat meningkatkannya produktivitas. Ketika robot berinteraksi dengan dunia yang kaya akan dan dunia yang bervariasi, ia menggunakan indera untuk mengumpulkan data dan kemudian membandingkan input sensorik dengan ekspektasi yang tertanam dalam model dunianya. Oleh karena itu, efektivitas robot dibatasi oleh keakuratannya pemrograman memodelkan dunia nyata

Dalam Perancangan kecerdasan buatan terdapat dua hal yang menjadi bagian penting yang dibutuhkan yaitu:

1. *Knowledge Base* (Basis pengetahuan) Pada bagian ini berisi tentang fakta – fakt, teori, pemikiran serta hubungan antara satu dengan yang lainnya.
2. *Inference Engine* (Motor inferensi) Berbeda dengan *Knowledge Base*, pada bagian *Inference Engine* (Motor inferensi) ini merupakan suatu kemampuan dalam menarik kesimpulan berdasarkan pengalaman.



**Gambar 2.3 Bagian utama Artificial Intelligence (AI)**

Menurut Rich dan Knight (1991), *Artificial Intelligence* (AI) adalah sebuah studi yang mempelajari tentang bagaimana membuat sebuah komputer melakukan hal – hal yang pada saat ini dapat dilakukan oleh manusia. Pada awal diciptakannya sebuah komputer, fungsi komputer hanya sebagai alat perhitungan saja. Namun, seiring dengan perkembangan zaman, peran komputer semakin mendominasi terhadap kehidupan manusia. Dimana komputer tidak hanya dijadikan sebagai alat hitung saja, namun diharapkan komputer agar mampu mengerjakan segala sesuatu yang mampu dikerjakan oleh manusia. Berikut adalah komponen utama dan jenis-jenis Artificial Intelligence/AI :

**a) Machine Learning**

*Machine Learning* muncul ketika pemikiran bagaimana teknologi yang ada dapat membantu manusia dalam memenuhi kebutuhan sehari-hari, munculnya pemikiran tersebut membuat *Machine Learning* sangat dibutuhkan karena dengan kehadirannya di cabang ilmu Kecerdasan Buatan (*Artificial Intelligence/AI*) dapat membantu manusia untuk memenuhi kebutuhan-kebutuhannya (Inkreswari Retno Hardini, 2019).

*Machine Learning* sendiri berfokus pada pengembangan sistem komputer yang mampu belajar dari data, mengidentifikasi pola, dan membuat keputusan dengan sedikit atau tanpa intervensi manusia langsung. *Machine Learning* mengacu

pada kemampuan komputer untuk belajar dari pengalaman (data) tanpa perlu secara eksplisit diprogram.

#### **b) Deep Learning**

*Deep learning* (DL) memiliki kaitan erat dengan jaringan saraf tiruan (*Neural Networks*), yang meniru struktur saraf dalam otak manusia. Pada awalnya, tujuannya adalah untuk mereplikasi proses kerja otak manusia. Namun, *Deep Learning* tidak hanya berupaya meniru operasi otak manusia, tetapi juga berusaha untuk mengembangkan jaringan saraf tiruan tersebut. Meskipun tidak akan secara sempurna menyerupai otak manusia dan operasinya, prinsip kerjanya sangat serupa (Zamachsari & Puspitasari, 2021).

### **2.4 Pengolahan Citra (Image Processing)**

Pengolahan citra atau image processing merupakan salah satu jenis teknologi yang digunakan untuk menyelesaikan masalah mengenai pemrosesan suatu gambar atau citra. Dalam pengolahan citra gambar akan di olah sedemikian rupa sehingga menghasilkan output-output yang akan digunakan untuk kebutuhan pengelompokan data. Data-data tersebut akan digunakan untuk keperluan lebih lanjut seperti mengklasifikasi dan mengidentifikasi (Fitri Muwardi, 2017). Citra juga dapat dikelompokkan menjadi citra tampak dan citra tidak tampak. Contoh citra tampak dalam kehidupan sehari-hari: foto, gambar, dan lukisan, sedangkan citra tidak tampak misalnya: data gambar dalam file (citra digital), dan citra yang direpresentasikan akan menjadi fungsi matematis (Gazali, 2012).

Untuk diimplementasikan sebuah citra harus melewati beberapa proses pengolahan, pengolahan citra digital bertujuan untuk menguba atau meningkatkan

kualitas citra sehingga lebih mudah untuk mendeteksi variabel-variabel tertentu. Dalam pengolahan citra digital, ada 3 tahap utama yang penting

1. Konversi Warna:

Langkah ini bertujuan untuk mengubah representasi warna citra ke dalam komponen warna lain agar lebih mudah dideteksi dan dianalisis. Misalnya, citra berwarna mungkin dikonversi ke skala abu-abu atau model warna lainnya untuk menyoroti fitur-fitur tertentu yang mungkin tidak terlihat jelas dalam format aslinya.

2. Segmentasi

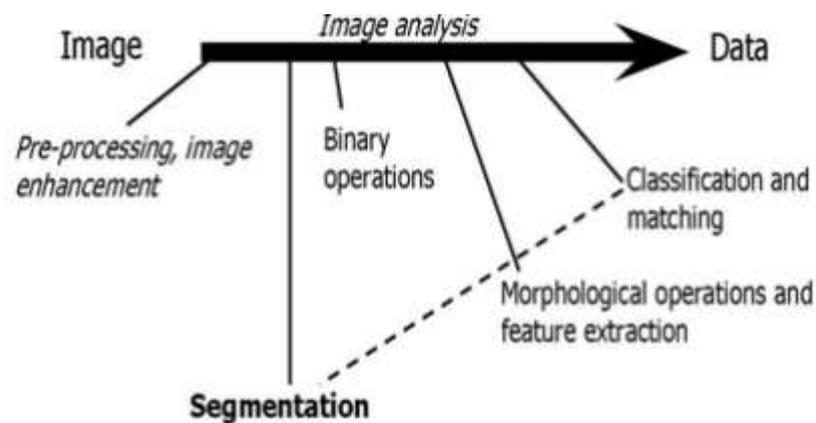
Segmentasi bertujuan untuk mengidentifikasi dan memisahkan bagian-bagian citra yang relevan, seperti lokasi, area, atau tepi objek yang dicari atau akan dianalisis. Proses ini membagi citra menjadi beberapa segmen atau wilayah yang homogen berdasarkan kesamaan karakteristik, seperti warna, intensitas, atau tekstur, sehingga objek yang diinginkan dapat diisolasi dari latar belakang.

3. Filter

Langkah ini bertujuan untuk membersihkan citra digital dari derau atau noise yang tidak diinginkan. Filter digunakan untuk menghaluskan citra, menghilangkan artefak yang mengganggu, dan meningkatkan kualitas visual citra. Proses ini penting untuk memastikan bahwa analisis selanjutnya, seperti deteksi objek atau ekstraksi fitur, dilakukan pada citra yang bersih dan jelas (Randy Cahya Wihandika, 2021).

## 2.5 Segmentasi Gambar (Image Segmentation)

Segmentasi adalah proses memisahkan citra menjadi bagian-bagian yang diharapkan merepresentasikan objek-objek tersendiri. Proses ini melibatkan pemisahan piksel dalam citra berdasarkan kesamaan tingkat keabuan atau karakteristik lainnya, sehingga piksel yang memiliki derajat keabuan serupa dikelompokkan bersama. Pada dasarnya, segmentasi bertujuan untuk memudahkan analisis citra dengan mengidentifikasi dan memisahkan area-area penting atau objek-objek dalam gambar. Teknologi modern juga sering menggunakan algoritma machine learning dan deep learning untuk segmentasi yang lebih akurat dan efisien (Tria Adhi Wijaya, 2010). Cara kerja segmentasi gambar dapat di lihat pada gambar 2.3.



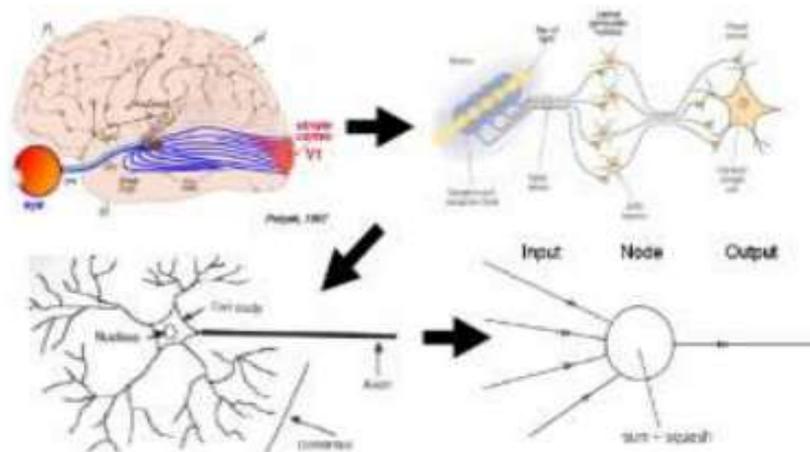
**Gambar 2.4** Cara kerja segmentasi gambar

## 2.6 Jaringan Syaraf Tiruan (Artificial Neural Network)

Jaringan syaraf tiruan (*Neural Network*) merupakan model pemrosesan informasi yang terinspirasi oleh sistem syaraf biologis yang terdapat dalam otak manusia. Sama halnya dengan cara otak memproses informasi melalui jaringan

neuron yang kompleks dan saling berhubungan, Aspek mendasar dari model ini adalah strukturnya yang khas untuk pemrosesan informasi.

Jaringan syaraf tiruan (JST) beroperasi dengan cara yang mirip dengan otak manusia, di mana mereka belajar dari contoh. Kemampuan belajar ini memungkinkan mereka untuk mengatasi masalah-masalah spesifik, seperti pengenalan pola dan klasifikasi. Melalui penerapan data pelatihan, JST mampu mengidentifikasi pola dan membuat keputusan, sehingga meningkatkan kinerjanya dari waktu ke waktu. Jaringan ini dibangun dengan lapisan-lapisan node yang saling terhubung, atau neuron buatan, yang secara kolektif memproses data input, mengekstrak fitur, dan menghasilkan output. Proses pembelajaran memerlukan penyesuaian bobot koneksi antar neuron berdasarkan perbedaan antara output aktual dan prediksi, sehingga memungkinkan jaringan untuk menyempurnakan prediksi dan klasifikasinya. Struktur dasar jaringan syaraf tiruan dapat dilihat pada gambar 2.5.



**Gambar 2.5 Struktur dasar jaringan syaraf tiruan**

Jaringan neuron buatan terdiri dari sekelompok neuron yang disusun dalam beberapa lapisan, berikut lapisan-lapisan nya:

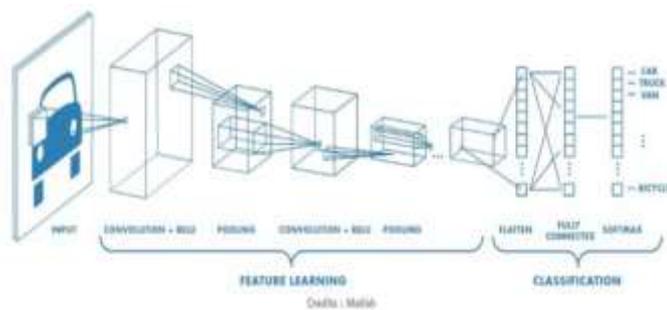
1. Lapisan input (Input Layer) adalah lapisan pertama dan bertanggung jawab untuk menerima dan memproses data input. Lapisan ini berfungsi sebagai penghubung antara jaringan dan sumber data eksternal.
2. Lapisan tersembunyi (hidden Layer) adalah subdivisi lebih lanjut dari jaringan, yang tidak dapat diakses oleh sumber data eksternal. Jaringan dapat terdiri dari satu atau lebih lapisan tersembunyi, atau mungkin tidak memiliki lapisan tersebut. Lapisan ini bertugas untuk memproses dan mentransformasi input yang diterima dari lapisan sebelumnya.
3. Lapisan output (Output Layer) bertanggung jawab untuk pemrosesan akhir dan transformasi input yang diterima dari lapisan sebelumnya. Neuron-neuron pada lapisan ini beroperasi dengan cara yang serupa dengan neuron-neuron pada lapisan tersembunyi. Fungsi sigmoid sering digunakan untuk menghasilkan output akhir dari proses jaringan (Fachrudin Pakaja, 2012).

Karena kapasitasnya untuk belajar dan beradaptasi, JST telah menjadi alat yang sangat berharga di berbagai bidang, termasuk pengenalan gambar dan ucapan, pemrosesan bahasa alami, dan analitik prediktif. Mereka mampu menangani kumpulan data yang besar dan kompleks, yang membuatnya sangat diperlukan dalam aplikasi berbasis data modern. Evolusi arsitektur jaringan saraf dan algoritme pelatihan yang sedang berlangsung semakin meningkatkan efektivitas dan memperluas potensi aplikasi sistem ini.

## **2.7 Convolutional Neural Network (CNN)**

*Convolutional Neural Network* (CNN) adalah jenis *Multi Layer Perceptron* (MLP) yang dirancang khusus untuk menganalisis gambar dua dimensi. CNN

meniru cara kerja otak manusia dalam mengenali objek yang terlihat, memungkinkan komputer untuk membedakan berbagai objek yang ada. Fitur ini dikenal sebagai pengenalan gambar. CNN merupakan metode klasifikasi dalam deep learning yang menggunakan lapisan konvolusi untuk memproses input dengan filter. Secara keseluruhan, CNN tidak jauh berbeda dari jaringan saraf biasa yang terdiri dari neuron dengan bobot, bias, dan fungsi aktivasi (Nihayatul Husna, 2022). Arsitektur CNN dapat dilihat pada Gambar 2.6.

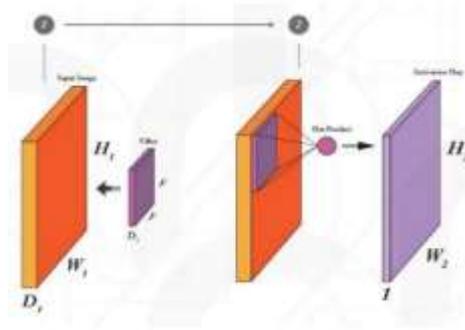


**Gambar 2.6 Arsitektur CNN**

Berikut ini penjelasan dari arsitektur di atas :

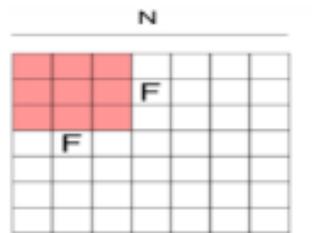
### 1. Convolutional Layer

*Convolutional layer* lapisan utama dan paling penting pada metode CNN dan menghasilkan citra baru yang menunjukkan fitur dari citra masukan. Convolutional layer terdiri dari neuron yang tersusun sehingga membentuk filter dengan panjang dan tinggi. (Anissa Ollivia Cahya Pratiwi, 2023). Berikut gambar convolutional layer dapat dilihat pada Gambar 2.7.



**Gambar 2.7 Convolutional Layer**

Pada setiap posisi gambar akan menghasilkan sebuah nilai yang berupa perkalian titik antara bagian gambar dengan filter yang digunakan. Berikut adalah ilustrasi proses Convolutional Layer.



**Gambar 2.8 Proses Convolutional Layer**

Ukuran spasial dari proses output gambar di atas, dihitung menggunakan rumus:

$$\frac{N - F + 2P}{S} + 1$$

Keterangan:

N: Ukuran spasial (tinggi H1 = lebar W1) dari input yang ada pada gambar.

F: Merupakan ukuran spasial filter

P: Suatu parameter di mana parameter ini menentukan jumlah pixel, lalu akan ditambahkan pada setiap sisi dari input, disebut juga sebagai Zero Padding

S: Besaran pergeseran filter pada setiap proses komputasi disebut juga dengan Stride.

Jika dilihat dari ilustrasi gambar di atas maka perhitungan komputasinya adalah sebagai berikut.

- $N = 7$
- $F = 3$
- $P$  (Padding) = 0
- $S$  (Stride) = 2 (Didapatkan berdasarkan pergeseran filter)

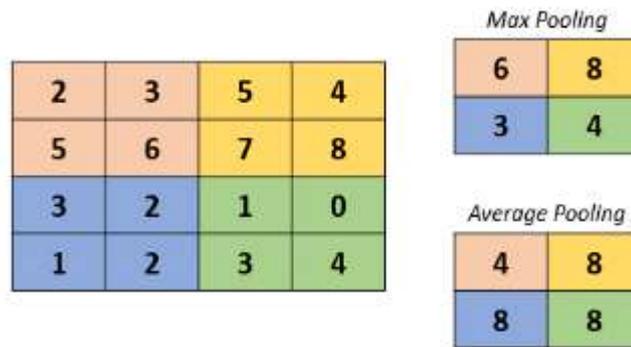
$$\text{Stride 1} \rightarrow \frac{7-3+0}{1} + 1 = 5$$

$$\text{Stride 2} \rightarrow \frac{7-3+0}{1} + 2 = 3$$

Proses komputasi ini akan diulang beberapa kali di setiap filter yang berbeda sampai menghasilkan beberapa kumpulan dari activation maps yaitu sebuah gambar baru. (Muhammad Rizky Firdaus, 2020).

## 2. *Pooling Layer* (Max Pooling)

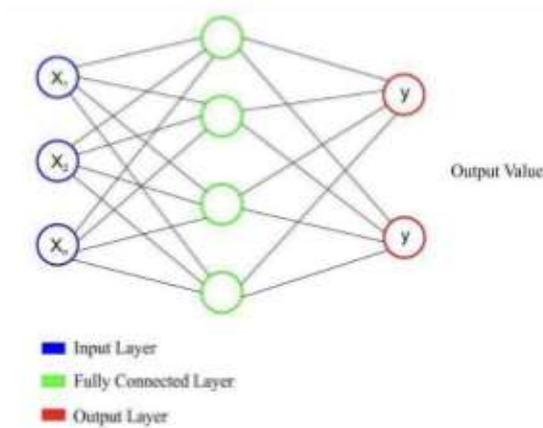
*Pooling layer* merupakan suatu layer yang digunakan untuk memperkecil ukuran dimensi dari feature map yang nantinya akan membantu untuk mempercepat Proses komputasi. Terdapat dua macam pooling yang sering digunakan yaitu Max Pooling dan Average Pooling. Jika max pooling mengambil nilai maksimal dari setiap piksel citra, maka average pooling mengambil nilai rata-rata. (Jurnal Fourier, 2023). Ilustrasi pooling layer ditunjukkan oleh Gambar 2.9.



**Gambar 2.9 Ilustrasi Pooling Layer**

### 3. Fully Connected Layer

*Fully connected layer* lapisan aktivitas neuron yang terhubung dengan lapisan neuron selanjutnya seperti jaringan syaraf tiruan. Setiap aktivitas neuron perlu diubah menjadi satu data dimensi untuk dapat dihubungkan ke semua neuron pada lapisan *Fully connected*. (Anissa Ollivia Cahya Pratiwi, 2023). Lapisan neuron Hasil feature map dari kedua layer sebelumnya masih berupa array multidimensi, sehingga perlu dilakukan proses reshape untuk mengubahnya menjadi vektor. Layer ini digunakan untuk mentransformasi data berdimensi agar dapat diklasifikasikan secara linear. Neuron-neuron pada convolutional layer harus diubah menjadi array satu dimensi sebelum dimasukkan ke dalam arsitektur ini. Proses ini bertujuan agar data tidak kehilangan informasi spasial dan tidak dapat dikembalikan ke bentuk semula. Arsitektur ini harus diimplementasikan pada akhir jaringan. (Muhammad Rizky Firdaus, 2020). Berikut gambar Fully Connected Layer, yang dapat dilihat pada Gambar 3.4.



**Gambar 2.10 Fully Connected Layer**

## 2.8 Tensorflow

Tensorflow merupakan platform open-source yang digunakan untuk pembelajaran mesin (Machine Learning) dan Deep Learning. Pada platform ini disediakan alat dan library yang luas untuk merancang, membangun, dan melatih model model Machine Learning seperti jaringan saraf tiruan. Tensorflow juga dirancang untuk menangani berbagai tugas dalam Machine Learning dan Deep Learning, termasuk klasifikasi, identifikasi, regresi, clustering, pengolahan citra (Pipit Dewi Arnesia, 2022).



**Gambar 2.11 Logo Tensorflow**

## **2.9 Keras**

Keras adalah API jaringan saraf tingkat tinggi yang ditulis dalam bahasa Python yang dapat berjalan di atas TensorFlow dan kerangka kerja tingkat rendah lainnya. API ini dirancang dengan fokus untuk memungkinkan eksperimen yang cepat dan memungkinkan prototipe (melalui keramahan pengguna, modularitas, dan ekstensibilitas) (melalui keramahan pengguna, modularitas, dan dapat diperluas). Keras menerima berbagai bentuk komponen jaringan saraf, seperti lapisan tebal, convolutional lapisan, lapisan berulang, lapisan putus, dan mendukung variasi-variasinya. Kode ini secara dinamis menangani alat-alat seperti Central Processing Unit (CPU) dan Graphics Processing Unit (GPU), sehingga memungkinkan penggunaan yang optimal. Ia juga memiliki implementasi fungsi aktivasi 8, pengoptimal 9, rumus metrik 10, dan prosedur yang diperlukan untuk menangani pelatihan sesi pelatihan dengan mudah . Plugin Deep Learning mengintegrasikan fungsionalitas dari pustaka Keras, yang pada gilirannya mengintegrasikan fungsi dari TensorFlow ke dalam Python (Bahzad Taha Chicho, 2021).

## **2.10 Python**

Python merupakan salah satu bahasa pemrograman dari sekian banyak bahasa pemrograman seperti Java, LISP, PHP, dan Perl. Python memiliki sifat yang dinamis, interpretatif, mudah di pelajari, dan memiliki sintaksis yang jelas dan mudah di baca, python juga sebagai bahasa pemrograman yang kuat dan serba guna yang dapat digunakan di berbagai bidang ilmu komputer dan pengembangan

perangkat lunak contohnya seperti pengolahan citra. Keunikan dan fleksibilitasnya membuat python sangat disukai para developer di seluruh dunia. Meskipun pada awalnya Python tidak mendapatkan perhatian sebesar bahasa pemrograman lainnya, kini Python telah menjadi sangat populer, terutama di bidang pembelajaran mesin dan kecerdasan buatan (AI). Python membuat pengembangan di bidang ini jauh lebih produktif dan lebih mudah, berkat pustaka yang kuat dan beragam, seperti TensorFlow, Keras, dan Scikit-Learn (Ajay Rawat, 2020).



**Gambar 2.13 Logo Python**

### **2.11 Google Colaboratory**

Google Colaboratory atau Google Colab adalah platform berbasis cloud untuk menulis, menjalankan, dan berbagi kode Python melalui web browser. Platform ini dirancang bagi analyst, developer, peneliti, dan pendidik yang bekerja di bidang data science dan machine learning dengan menyediakan environment komputasi yang fleksibel dan mudah diakses tanpa biaya. Google Colab juga menawarkan kemampuan untuk menjalankan Jupyter Notebook (web app open-source untuk kombinasi kode, teks terformat, dan visualisasi data) langsung dari web browser tanpa perlu konfigurasi apa pun (Febrywinata, E, 2024).

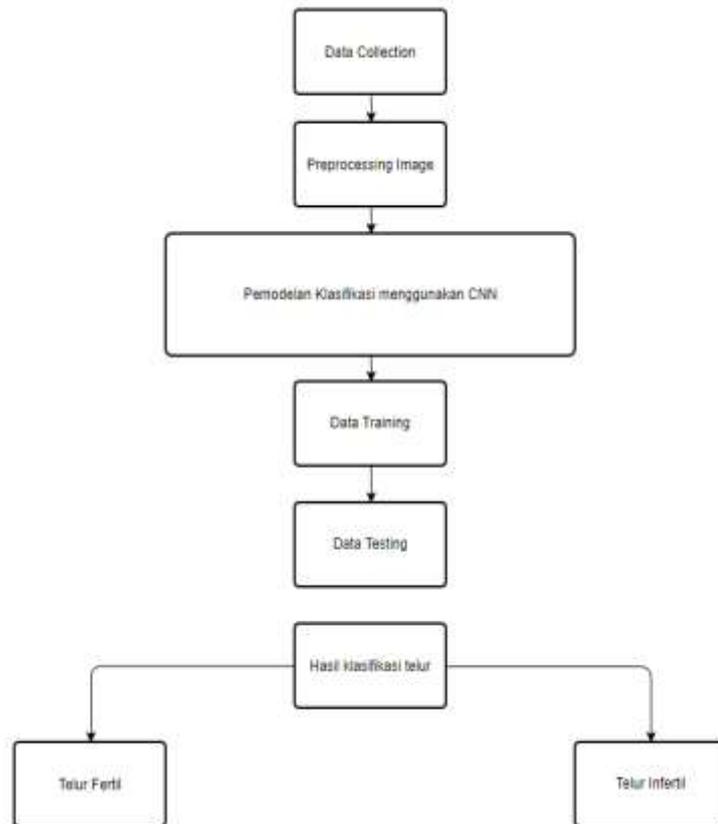
## **BAB III**

### **METODOLOGI PENELITIAN**

#### **3.1 Gambaran Umum Objek Penelitian**

Penelitian ini berfokus pada mengklasifikasi perbedaan jenis telur fertil dan infertil menggunakan metode *Convolutional Neural Network* (CNN) dengan memanfaatkan library Tensorflow. Tujuan utama dari penelitian ini adalah mengklasifikasi secara akurat jenis telur ayam fertil dan infertil dengan menggunakan jaringan syaraf tiruan yang memiliki keefektivitas yang tinggi.

Pengembangan sistem mengklasifikasi telur ini terdiri dari beberapa langkah penting, dimulai dengan pengumpulan dan pemrosesan data awal, merancang metode CNN, serta tahap pengujian dan klasifikasi. Setiap tahapan yang dilakukan dengan cermat dan sistematis untuk menghasilkan hasil klasifikasi yang akurat dan optimal. Dengan pendekatan ini diharapkan sistem yang dihasilkan tidak hanya berfungsi dengan baik untuk mengklasifikasi telur fertil dan infertil tapi juga memberikan cara yang lebih efisien untuk membedakan jenis telur bagi para pengguna. Teknik perancangan sistem dapat dilihat pada gambar 3.1.

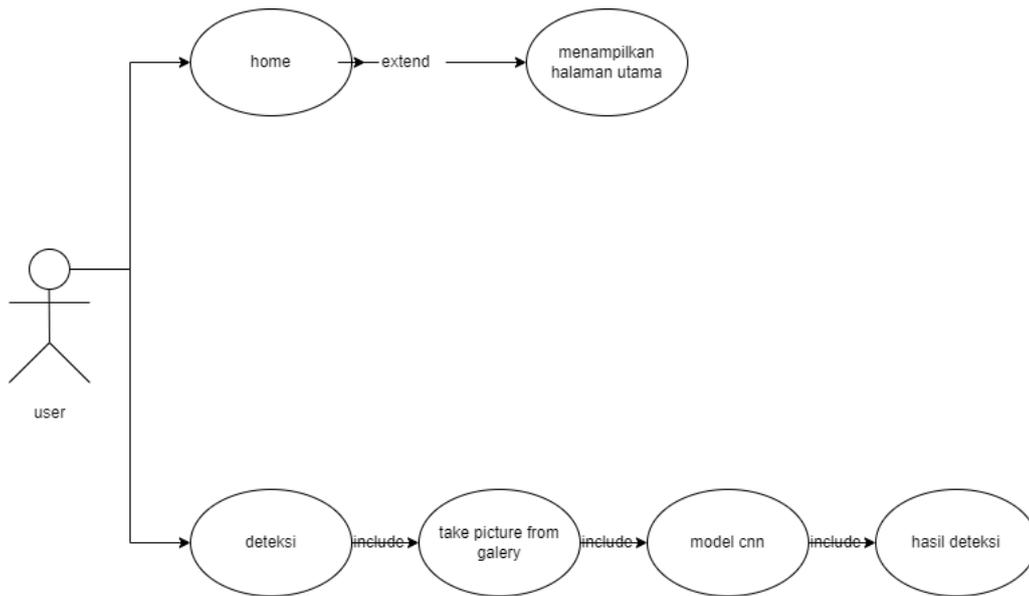


**Gambar 3.1 Teknik perancangan sistem**

### 3.2 UML (Unified Modelling Language)

#### Use Case Diagram

Dalam Gambar 3.2 dibawah ini terdapat satu kategori pengguna, yaitu user. Dibawah ini merupakan penjelasan mengenai desain diagram use case ketika user masuk keaplikasi sistem akan menampilkan halaman awal atau beranda. Yang dimana pada halaman beranda tersebut terdapat beberapa fitur menu yang bisa diakses.

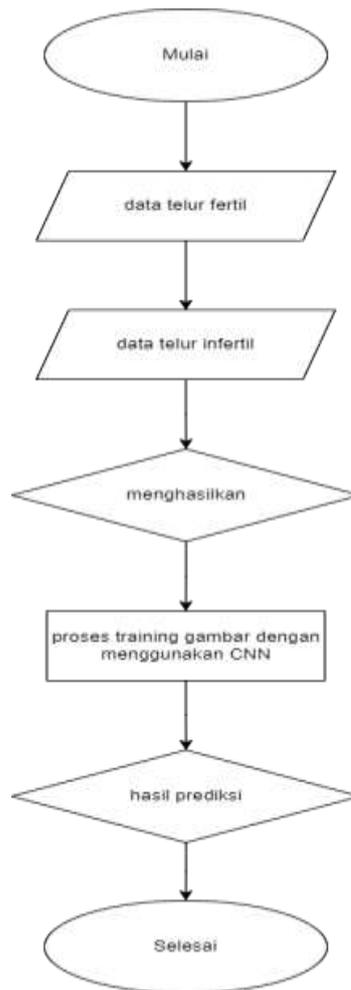


**Gambar 3.2 Use Case Diagram**

### **3.3 Konsep Pemodelan Sistem**

#### **Flowchart**

Dalam Gambar 3.3 dibawah ini merupakan flowchart sistem, dimulai dari login selanjutnya impor gambar melalui kamera atau upload gambar dari galeri. Setelah itu sistem mendeteksi jenis telur dan sistem melakukan perhitungan algoritma cnn. Setelah itu menampilkan hasil diagnosa dari proses tersebut.



**Gambar 3.3 Flowchart Sistem**

### **3.4 Data Collection**

Pengumpulan data merupakan aspek yang sangat penting dalam penelitian ini, khususnya dalam rangka membangun sistem klasifikasi telur fertil dan infertil. Tahap ini sangat krusial karena kualitas dan variasi data yang dikumpulkan akan sangat mempengaruhi akurasi dalam klasifikasi telur fertil dan infertil. Data set berupa citra atau gambar dari telur fertil dan infertil yang di peroleh dari 2 sumber utama.

1. Pertama, data diambil dari pengambilan gambar telur fertil dan infertil langsung di lapangan yang dimana tujuan dari teknik pengambilan data langsung di lapangan untuk mendapatkan informasi yang lebih spesifik dari hasil langsung telur fertil dan infertil. Sehingga sebagai penguji dapat membedakan mana telur fertil dan infertil yang lebih spesifik.
2. Kedua, data diambil dari opensource seperti google image dll, data yang di kumpulkan disini hanya untuk menunjang atau menutupi data yang kurang dari pengambilan di lapangan sehingga data dapat mencukupi atau tercukupi ketika berada di tahapan preprocessing atau data training. Pengambilan data secara open source ini juga di harapkan dapat digunakan ketika dalam proses klasifikasi menggunakan Algoritma *Convolutional Neural Network* (CNN).

Ukuran Gambar	Pembagian Data	
256 x 256	Telur	
	Telur Fertil	Telur Infertil
	50	50

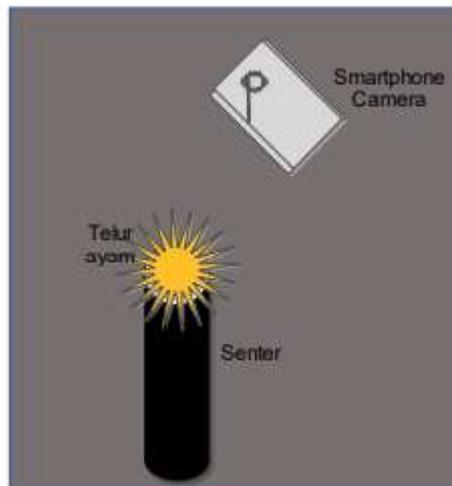
**Tabel 3.1 Tabel data set**

### **3.5 Preprocessing Data**

Tahap Preprocessing, pada tahapan ini semua data yang di peroleh akan di kumpulkan menjadi 1 dan kemudian akan di bedakan menjadi 2 jenis data, yaitu data telur fertil dan data telur infertil. Tujuan dari Preprocessing Data disini untuk memudahkan data untuk nanti di training dan di uji. Data-data yang di kumpulkan juga melewati tahapan penyesuain gambar seperti ukuran panjang (height), lebar

(width) dan ukuran (size) sehingga seluruh data set mempunyai format dan ukuran yang sama tujuannya adalah untuk memudahkan dalam tahap proses Data Testing, semua ukuran gambar nantinya akan disesuaikan menjadi 256 piksel.

Proses awalnya pada tahapan ini adalah akusisi, akusisi dilakukan dengan menggunakan senter untuk penyinaran telur, penyinaran telur sendiri berfungsi untuk melihat sel sel yang ada pada dalam telur yang tidak dapat dilihat dengan mata dan diambil dengan menggunakan kamera smartphone untuk proses pengambilan gambar berupa citra warna (RGB) dengan jumlah data set sebesar data (25 data telur fertil dan 25 data telur infertil). Teknik pengambilan data set telur fertil dan infertil seperti pada gambar 3.5.

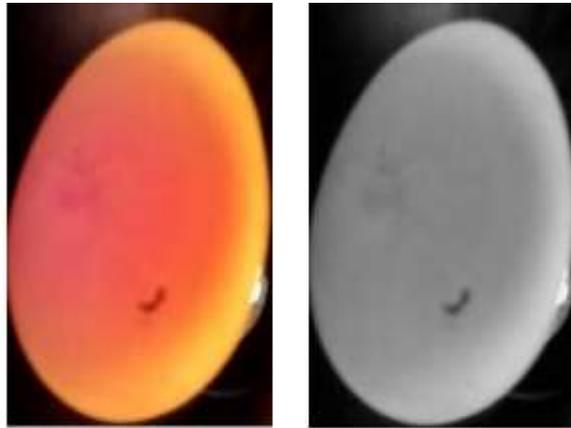


**Gambar 3.5 Teknik pengambilan data set telur**

### **3.6 Grayscale**

Pada tahap ini hasil proses dari akusisi citra akan lanjut di lakukan preprocessing dengan menggunakan metode grayscale, Proses ini dilakukan pada data set telur dengan menggunakan nilai matriks dari citra warna dengan masing-masing komponen warna (R,G,dan B) dimasukkan ke dalam persamaan 1 akan

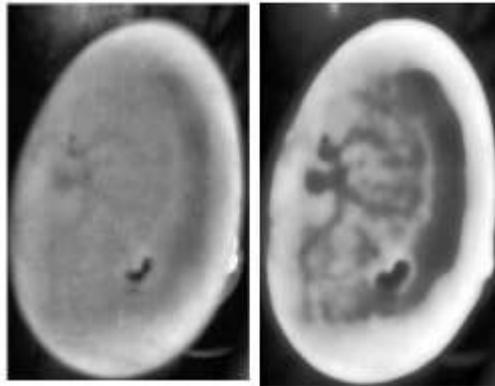
menghasilkan citra grayscale (citra aras keabuan dengan intensitas nilai dengan range 0 (hitam)-255 (putih). Hasil proses grayscale pada telur dapat dilihat pada gambar 3.4.



**Gambar 3.4 Proses Grayscale pada telur**

### **3.7 Image Enchanment**

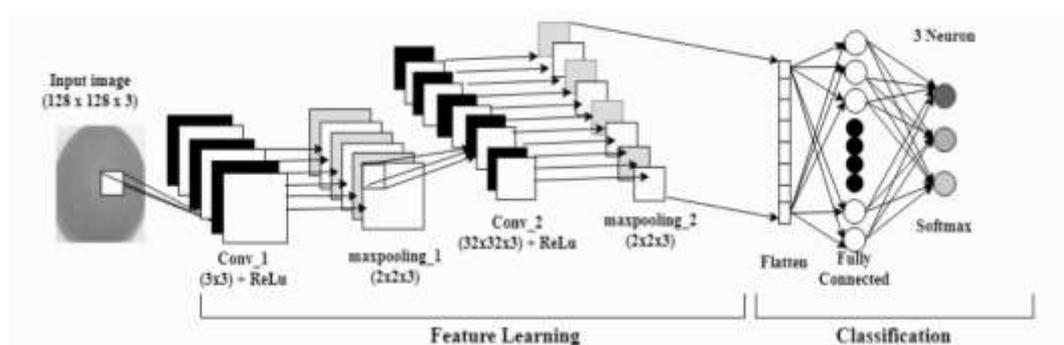
Setelah proses grayscale dilakukan selanjutnya data set telur yang telah melewati fase grayscale akan dilakukan image enchanment dengan menggunakan metode gabungan Clahe (Contrast Limited Adaptive Histogram Equalization) dengan He (Histogram Equalization). Proses gabungan clahe dengan he akan menampilkan data gambar yang lebih kontras di dalam telur terlihat lebih jelas jika dibandingkan dengan citra hasil grayscale. Perbandingan citra Grayscale dengan Clahe dan He dapat dilihat pada gambar 3.3



**Gambar 3.6 Perbandingan citra Grayscale dengan Clahe dan He**

### 3.8 Pemodelan Klasifikasi Menggunakan CNN

Pada Proses ini, sistem akan melatih data set telur menggunakan algoritma *Convolutional Neural Network* (CNN). Input seluruh data set telur yang digunakan berukuran 256x256 pixel, tujuan yaitu untuk mendapatkan hasil evaluasi dan pemodelan yang optimal, maka pada penelitian ini menggunakan aktivasi ReLu, dua pooling layer dan fully connected layer. Berikut gambaran rancangan arsitektur dengan algoritma *Convolutional Neural Network* (CNN) dalam penelitian ini:



**Gambar 3.7 Arsitektur CNN**

Berikut ini adalah penjelasan tentang cara kerja sistem arsitektur CNN di atas :

1. Gambar yang digunakan sebagai input adalah hasil proses pembesaran dengan ukuran 256x256 piksel. Ukuran 256x256x3 menunjukkan bahwa gambar tersebut memiliki dimensi 256x256 piksel, sedangkan angka 3 menunjukkan jumlah kanal gambar, yang berarti bahwa gambar tersebut adalah gambar RGB (merah, hijau dan biru).
2. Kemudian masuk ke proses konvolusi pertama. Dalam proses ini, input gambar akan dikalikan dengan kernel berukuran 3x3 dengan 32 filter. Proses perkalian dilakukan dengan menggeser kernel satu kali, menghasilkan map fitur 256 x 256. Setelah proses konvolusi selesai, ditambahkan fungsi aktivasi ReLu. Tujuan fungsi ini adalah untuk mengubah nilai negatif pada matriks yang menghasilkan proses konvolusi menjadi nol.
3. Proses pooling pertama akan mengurangi ukuran feature map yang dihasilkan dari proses konvolusi pertama. Proses ini akan menggunakan teknik max-pooling, yang menggunakan kernel berukuran 2x2. Teknik ini menghitung nilai tertinggi berdasarkan pergeseran kernel sebanyak nilai stride, yaitu 2. Dengan demikian, proses pooling pertama akan menghasilkan matriks baru berukuran 63 x 63 piksel.
4. Kemudian, dalam proses konvolusi kedua, input feature map yang diperoleh dari konvolusi pertama dikalikan silang dengan kernel berukuran 3 kali 3 dengan filter sebanyak 32. Seperti halnya dalam konvolusi pertama, dalam proses konvolusi kedua juga dilakukan perhitungan aktivasi fungsi ReLu.
5. Proses pooling kedua, yang hampir sama dengan proses pooling pertama, menggunakan teknik maxpooling untuk mengurangi ukuran peta fitur dengan

kernel berukuran 2 kali 2 dengan stride 2 piksel. Hasil dari proses pooling kedua adalah peta fitur berukuran 31 kali 31 piksel.

6. Setelah tahap pooling kedua, proses flatten digunakan untuk mengubah map fitur yang terdiri dari array 3D menjadi vector 1D, yang digunakan sebagai inputan pada proses lapisan yang fully connected layer.
7. Selanjutnya, vektor yang dihasilkan dari proses flatten akan dihubungkan ke beberapa lapisan penuh terhubung melalui lapisan fully-connected. Pada saat ini, hanya ada satu lapisan yang tersembunyi di jaringan yang digunakan.

Sebelum klasifikasi gambar dimulai, digunakan nilai Dropout sebesar 0,5 untuk mengurangi tingkat overfitting selama proses pelatihan.

8. Proses terakhir menggunakan fungsi softmax untuk melakukan proses klasifikasi dan menghasilkan jumlah kelas yang sesuai dengan kategori yang telah ditentukan.

Proses penulisan kode dilakukan menggunakan Google Colab, platform yang didedikasikan untuk kebutuhan pengembangan pembelajaran mesin. Sebelum mulai menggunakan *Convolutional Neural Network* untuk mengklasifikasikan telur fertil dan infertil, Anda harus menggunakan beberapa library sebelum memulai. Tensorflow, juga dikenal sebagai TF, adalah library *opensource* yang digunakan untuk pengembangan *machine learning*. TensorFlow akan membagi komputasi graf sebelumnya dan menjalankannya secara bersamaan. Pelatihan model jaringan saraf akan lebih mudah dengan Tensorflow ini. Lalu library Keras, library ini merupakan library yang dibangun di atas backend TensorFlow yang berfungsi untuk membuat sebuah *image classifier*. Dengan

library Keras proses membangun jaringan saraf tiruan untuk klasifikasi gambar akan menjadi lebih mudah. Penggunaan dari library tersebut dilakukan dengan cara mengimpor pada header.

### **3.9 Data Training**

Pada tahap ini masuk pada tahapan data training, tahap training data dengan metode CNN ini merupakan tahapan yang sangat penting karena training yang baik akan membantu tingkat keakuratan sistem permodelan dalam mengklasifikasi telur ayam fertil dan telur ayam infertil. Pada proses data training ini data set yang telah di kumpulkan yang berasal dari 2 sumber utama yang berupa citra atau gambar nantinya akan digunakan untuk melatih model klasifikasi telur ayam fertil dan telur ayam infertil, dengan menggunakan sistem arsitektur CNN dapat keefektifan dan tingkat akurasi sangat membantu dalam mengenali pola visual dan fitur dari data set telur yang berupa citra atau gambar.

Proses training data melibatkan beberapa eksperimen dengan berbagai parameter, seperti jumlah epoch, ukuran batch, dan nilai learning rate, untuk menemukan kombinasi terbaik yang menghasilkan akurasi optimal. Dengan menggunakan framework Keras dan backend TensorFlow, pelatihan dilakukan dalam lingkungan Vscod menggunakan bahasa pemrograman Python. Hasil dari pelatihan ini diharapkan mampu membedakan dengan akurasi tinggi antara telur fertil dan infertil berdasarkan karakteristik visual yang telah dipelajari oleh model CNN.

<b>Jenis Data</b>	<b>Telur Fertil</b>	<b>Telur Infertil</b>
Data Asli	50 Data	50 Data
Data Training	40 Data	40 Data

**Tabel 3.2 Tabel data training**

### **3.10 Data Testing**

Pada proses pengujian ini menguji seberapa baik kinerja dari model atau data yang sudah di training. Pada proses testing ini sistem akan menguji data sebanyak 20 data dengan masin-masing kelas, tujuannya agar dapat di ketahui seberapa baik kinerja model yang telah dibuat untuk klasifikasi telur fertil dan infertil. Pada proses testing akan menghitung nilai loss dan akurasi dari model yang di hasilkan dari data uji, nantinya digunakan sebagai klasifikasi terhadap telur. Dengan demikian, dapat di ketahui hasil dari klasifikasi telur ayam berdasarkan bercak nokta darah di bagian dalam telur menggunakan algoritma CNN.

<b>Jenis Data</b>	<b>Telur Fertil</b>	<b>Telur Infertil</b>
Data Asli	50 Data	50 Data
Data Testing	10 Data	10 Data

**Tabel 3.3 Tabel data testing**

### **3.11 Evaluation Model**

Setelah model selesai dilatih dan dievaluasi, langkah terakhir dalam penelitian ini adalah menguji kinerja model menggunakan data uji yang sepenuhnya baru. Data uji sangat berperan penting dalam menilai kemampuan model dalam memprediksi data yang belum pernah di gunakan selama proses pelatihan dan

evaluasi. Dengan demikian, evaluasi ini memberikan penilaian objektif terhadap kinerja keseluruhan model yang telah dikembangkan. Langkah ini sangat penting untuk menilai keefektifitas sistem model dan memastikan bahwa model tidak hanya berfungsi dengan baik.

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

**4.1. Hasil Pengumpulan Data Set**

Pada penelitian ini menggunakan dataset telur yang terdiri dari dua kelas yaitu fertil dan infertil yang datasetnya didapatkan dari observasi atau foto langsung menggunakan kamera handphone.

No	Nama Kelas	Jumlah Dataset
1	Fertil	50 Dataset
2	Infertil	50 Dataset
Jumlah		100 Dataset

**Tabel 4.1 Dataset**

Berdasarkan tabel di atas, dataset pada kelas fertil berjumlah 50 dataset sedangkan kelas infertil berjumlah 50 dataset. Sehingga total keseluruhan dataset yang digunakan pada penelitian ini yaitu 100 dataset. Karena model akhir diharapkan mampu beradaptasi dengan keadaan nyata yang serupa dengan data pelatihan, maka penggunaan dataset langsung dari observasi lapangan menambah nilai pada penelitian ini. Setelah itu, model CNN yang dibangun dilatih untuk mengidentifikasi isyarat visual yang membedakan antara telur yang layak dan tidak subur, sehingga memungkinkannya membuat prediksi yang tepat ketika diterapkan pada gambar telur fertil dan infertil.



**Gambar 4.1 Dataset Telur Fertil**

Telur ayam fertil adalah telur yang telah dibuahi oleh sperma pejantan, memungkinkan mereka untuk berkembang menjadi embrio dan kemudian menetas menjadi anak ayam. Proses pembuahan ini mirip dengan proses reproduksi pada hewan lain, termasuk manusia, di mana sel telur betina bertemu dengan sel sperma jantan, menghasilkan zigot yang siap untuk berkembang. Setelah perkawinan, saluran reproduksi betina ayam memungkinkan pembuahan. Tidak seperti telur infertil atau telur yang tidak dibuahi, yang hanya terdiri dari sel telur tanpa perkembangan embrio, telur fertil akan memulai proses inkubasi ketika dibiarkan dalam kondisi yang sesuai seperti suhu yang hangat dan lembap. Pada tahap ini, sel-sel mulai membelah dan berkembang, membentuk embrio yang tumbuh hingga menetas.



**Gambar 4.2 Dataset Telur Infertil**

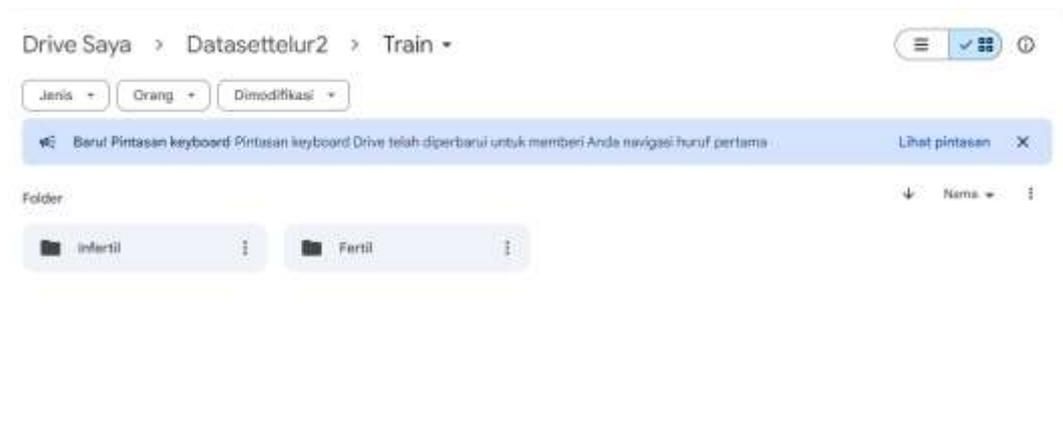
Telur infertil biasanya terjadi dalam kasus di mana pejantan tidak terlibat dalam proses reproduksi atau ketika sistem reproduksi pejantan atau betina mengalami masalah yang menghalangi pembuahan. Dengan kata lain, sperma pejantan tidak membuahi telur ini, sehingga mereka tidak dapat berkembang menjadi embrio atau menetas menjadi anak ayam. Tidak seperti proses reproduksi manusia, sel telur yang tidak dibuahi dikeluarkan dari tubuh selama siklus menstruasi daripada berkembang menjadi embrio; namun, telur infertil biasanya memiliki nutrisi yang sama seperti telur normal.

## **4.2. Implementasi Model CNN**

### **4.2.1. Persiapan Dataset**

Dataset yang digunakan dalam penelitian ini terdiri dari seratus dataset yang merupakan representasi dari data yang akan digunakan untuk pelatihan model. Dataset ini kemudian diunggah ke Google Drive, yang memungkinkan pengelolaan data yang fleksibel, terutama ketika bekerja di Google Colaboratory, sebuah

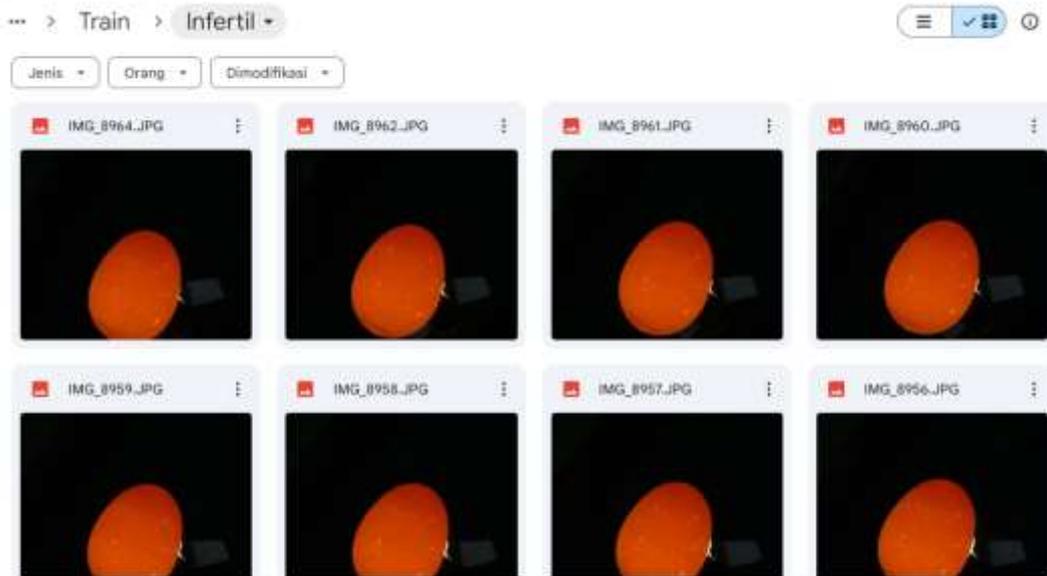
platform berbasis cloud yang populer. Proses pemanggilan dan penggunaan data menjadi lebih efisien dengan menyimpan dataset di Google Drive. Setelah pelatihan model dilakukan di Google Colaboratory, data dapat diakses langsung dari Google Drive tanpa mengunduhnya ke perangkat lokal.



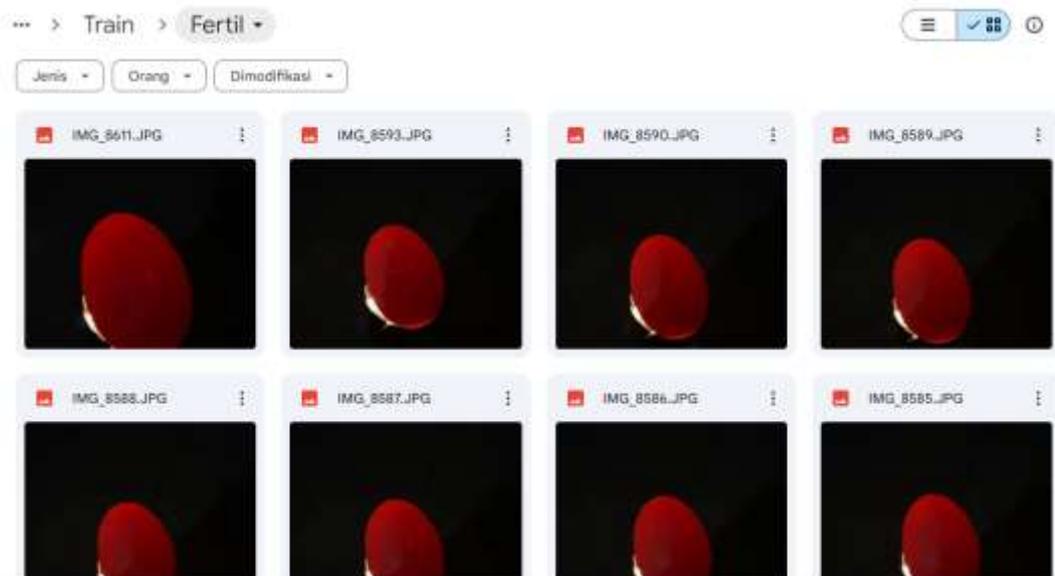
**Gambar 4.3 Folder Kelas**

Untuk memudahkan proses pelatihan model klasifikasi, dataset yang saya gunakan disimpan di Google Drive dalam struktur folder yang terorganisir. Dataset yang disebut sebagai "Datasettelur2" memiliki subfolder yang disebut "Train" yang berfungsi sebagai tempat penyimpanan data yang akan digunakan untuk melatih model. Dataset lebih lanjut diorganisir di dalam folder "Train" ke dalam dua subfolder: "Fertil" dan "Infertil". Folder "Fertil" berisi foto telur yang telah dibuahi oleh pejantan, yang mungkin menetas menjadi anak ayam, sedangkan folder "Infertil" berisi foto telur yang tidak dibuahi, sehingga tidak dapat berkembang menjadi embrio. Struktur folder ini dibuat untuk mempermudah proses pelatihan model. Dalam proses ini, masing-masing gambar dalam subfolder akan digunakan sebagai input untuk mengajarkan model untuk membedakan telur fertil dan infertil berdasarkan ciri-ciri visualnya. Dataset seperti ini juga membantu memastikan

bahwa data terlabel dengan benar dan meminimalkan kesalahan saat model memproses gambar selama pelatihan.



**Gambar 4.4 Infertil Pada Google Drive**



**Gambar 4.5 Fertil Pada Google Drive**

#### 4.2.2. Menghubungkan Google Drive

Tahap ini untuk menghubungkan Google Colab dengan Google Drive yang memungkinkan pengguna untuk mengakses data yang ada di Google Drive secara langsung dari Google Colab. Modul OS diimpor untuk menangani operasi sistem seperti mengelola path direktori. Mountpoint menunjukkan lokasi di mana Google Drive akan di-mount atau dihubungkan ke Google Colab. Kemudian datadir diaktifkan dengan fungsi `os.path.join` untuk menggabungkan mountpoint dengan path folder tertentu di Google Drive.

```
import os
from google.colab import drive
MOUNTPOINT =  '/content/gdrive'
DATADIR = os.path.join(MOUNTPOINT, 'Data', 'myfolder')
drive.mount(MOUNTPOINT)
```

**Gambar 4.6 Menghubungkan Google Drive**

#### 4.2.3. Import Library

Pada tahap kodingan ini menggunakan TensorFlow dan Keras, tahap ini melibatkan impor berbagai library dan modul yang diperlukan untuk membuat, melatih, dan menilai model jaringan saraf tiruan. Kode ini juga mengimpor alat untuk pemrosesan data, tampilan visualisasi, dan analisis hasil. TensorFlow dan Keras berfungsi sebagai kerangka kerja utama untuk membangun model, sedangkan NumPy, Pandas, dan Matplotlib adalah library yang digunakan untuk memproses data dan membuat visualisasi. Dataset dibagi menjadi data pengujian dan pelatihan dengan Scikit-learn. Selain itu, metrik seperti report klasifikasi dan confusion matrix digunakan untuk mengevaluasi kinerja model.

```
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import backend as K
from tensorflow.keras.layers import Dense, Activation, Dropout, Conv2D, MaxPooling2D, BatchNormalization, Flatten, Input
from tensorflow.keras.optimizers import Adam, Adamax
from tensorflow.keras.metrics import categorical_crossentropy
from tensorflow.keras import regularizers
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.models import Model, load_model, Sequential
from keras.callbacks import ModelCheckpoint
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt
from matplotlib.pyplot import imshow
import os
import seaborn as sns
sns.set_style("darkgrid")
from sklearn.metrics import confusion_matrix, classification_report
from IPython.display import display, HTML
```

**Gambar 4.7 Import Library**

#### **4.2.4. Import Arsitektur VGG16**

Pada tahap ini, model VGG16 terlatih diimpor, instance model dibuat, dan detail arsitektur model ditampilkan. Hal ini berguna untuk memahami arsitektur jaringan dan memverifikasi bahwa model memenuhi kriteria sebelum digunakan dalam proyek pembelajaran mendalam.

```
#import library yang di butuhkan
from keras.applications.vgg16 import VGG16

#deklarasi model yang akan di pakai
model = VGG16()

#tampilkan layer-layer model VGG16
model.summary()
```

Model: "vgg16"

Layer (type)	Output Shape	Param #
input_layer (InputLayer)	(None, 224, 224, 3)	0
block1_conv1 (Conv2D)	(None, 224, 224, 64)	1,792
block1_conv2 (Conv2D)	(None, 224, 224, 64)	36,928
block1_pool (MaxPooling2D)	(None, 112, 112, 64)	0
block2_conv1 (Conv2D)	(None, 112, 112, 128)	73,856
block2_conv2 (Conv2D)	(None, 112, 112, 128)	147,584
block2_pool (MaxPooling2D)	(None, 56, 56, 128)	0

**Gambar 4.8 Import VGG16**

#### 4.2.5. Pemeriksaan Direktori dan File Dataset

Pada tahap ini memastikan bahwa semua file dan direktori yang diperlukan untuk melatih model klasifikasi telur fertil dan infertil berada di lokasi yang tepat di Google Drive.

```
# !ls /content
# !ls /content/gdrive
# !ls /content/gdrive/MyDrive
!ls /content/gdrive/MyDrive/Datasettelur2/Train
!ls /content/gdrive/MyDrive/Datasettelur2/Train/Fertil
!ls /content/gdrive/MyDrive/Datasettelur2/Train/Infertil
```

Fertil Infertil

```
IMG_8358.JPG IMG_8446.JPG IMG_8454.JPG IMG_8561.JPG IMG_8574.JPG IMG_8583.JPG IMG_8593.JPG
IMG_8378.JPG IMG_8447.JPG IMG_8455.JPG IMG_8564.JPG IMG_8575.JPG IMG_8584.JPG IMG_8611.JPG
IMG_8412.JPG IMG_8448.JPG IMG_8456.JPG IMG_8567.JPG IMG_8576.JPG IMG_8585.JPG
IMG_8437.JPG IMG_8449.JPG IMG_8457.JPG IMG_8568.JPG IMG_8577.JPG IMG_8586.JPG
IMG_8442.JPG IMG_8450.JPG IMG_8458.JPG IMG_8569.JPG IMG_8579.JPG IMG_8587.JPG
IMG_8443.JPG IMG_8451.JPG IMG_8459.JPG IMG_8570.JPG IMG_8580.JPG IMG_8588.JPG
IMG_8444.JPG IMG_8452.JPG IMG_8460.JPG IMG_8572.JPG IMG_8581.JPG IMG_8589.JPG
IMG_8445.JPG IMG_8453.JPG IMG_8560.JPG IMG_8573.JPG IMG_8582.JPG IMG_8590.JPG
IMG_8774.JPG IMG_8880.JPG IMG_8890.JPG IMG_8898.JPG IMG_8946.JPG IMG_8954.JPG IMG_8962.JPG
IMG_8806.JPG IMG_8882.JPG IMG_8891.JPG IMG_8899.JPG IMG_8947.JPG IMG_8955.JPG IMG_8964.JPG
IMG_8873.JPG IMG_8883.JPG IMG_8892.JPG IMG_8900.JPG IMG_8948.JPG IMG_8956.JPG
IMG_8874.JPG IMG_8884.JPG IMG_8893.JPG IMG_8901.JPG IMG_8949.JPG IMG_8957.JPG
IMG_8875.JPG IMG_8885.JPG IMG_8894.JPG IMG_8905.JPG IMG_8950.JPG IMG_8958.JPG
```

**Gambar 4.9 Pemeriksaan Dataset**

#### 4.2.6. Visualisasi Dataset

Tahap ini untuk memuat dan menampilkan gambar dari dataset yang tersimpan di Google Drive.



Gambar 4.10 Visualisasi Dataset

#### 4.2.7. Dataframe Dataset

Tahap ini membahas bagaimana file gambar dan labelnya diambil dan disusun ke dalam dataframe. Proses ini kemudian dapat digunakan untuk mengajarkan model dalam proyek klasifikasi gambar.

```
sdir="/content/gdrive/MyDrive/Datasettelur2/Train"
filepaths=[]
labels=[]
classlist=os.listdir(sdir)
for class in classlist:
    classpath=os.path.join(sdir,class)
    if os.path.isdir(classpath):
        flist=os.listdir(classpath)
        for f in flist:
            fpath=os.path.join(classpath,f)
            filepaths.append(fpath)
            labels.append(class)
Fseries= pd.Series(filepaths, name='filepaths')
Lseries=pd.Series(labels, name='labels')
df=pd.concat([Fseries, Lseries], axis=1)
print (df.head())
print (df['labels'].value_counts())
```

	filepaths	labels
0	/content/gdrive/MyDrive/Datasettelur2/Train/In...	Infertil
1	/content/gdrive/MyDrive/Datasettelur2/Train/In...	Infertil
2	/content/gdrive/MyDrive/Datasettelur2/Train/In...	Infertil
3	/content/gdrive/MyDrive/Datasettelur2/Train/In...	Infertil
4	/content/gdrive/MyDrive/Datasettelur2/Train/In...	Infertil

Gambar 4.11 Dataframe Dataset

#### 4.2.8. Split Data

Pada tahapan ini dilakukan split data atau pembagian dataset ke dalam folder train, validation dan test. Dataset yang berjumlah 100 akan dibagi menjadi 40:40:20 yang berarti 40 dataset berada di train, 40 dataset berada di validation, dan 20 dataset berada di test.

```
train_split=.4
test_split=.2
dummy_split=test_split/(1-train_split)
train_df, dummy_df=train_test_split(df, train_size=train_split, shuffle=True, random_state=8)
test_df, valid_df=train_test_split(dummy_df, train_size=dummy_split, shuffle=True, random_state=8)
print ('train_df length: ', len(train_df), ' test_df length: ', len(test_df), ' valid_df length: ', len(valid_df))

train_df length: 40 test_df length: 20 valid_df length: 40
```

**Gambar 4.12 Split Data**

#### 4.2.9. Data Generator

Tujuan ini untuk mempersiapkan data generator untuk digunakan dalam proses pelatihan, validasi, dan pengujian model deep learning. Generator ini harus memuat gambar, mengubah ukurannya, dan menormalisasi nilainya sebelum diteruskan ke model untuk melatih atau menguji performa. Dimensi gambar diubah menjadi 150 x 150 piksel dan skala abu-abu (grayscale) dengan satu channel.

```

height = 150
width = 150
channels = 1 # Grayscale has 1 channel
batch_size = 64

img_shape = (height, width, channels)
img_size = (height, width)
length = len(test_df)
test_batch_size = sorted([int(length/n) for n in range(1,length+1) if length % n == 0 and length/n <= 80], reverse=True)[0]
test_steps = int(length/test_batch_size)
print('Test batch size:', test_batch_size, ' Test steps:', test_steps)

# Data generators
gen = ImageDataGenerator(rescale=1./255)

train_gen = gen.flow_from_dataframe(
    train_df,
    x_col='filepaths',
    y_col='labels',
    target_size=img_size,
    class_mode='categorical',
    color_mode='grayscale', # Convert images to grayscale
    shuffle=True,
    batch_size=batch_size
)

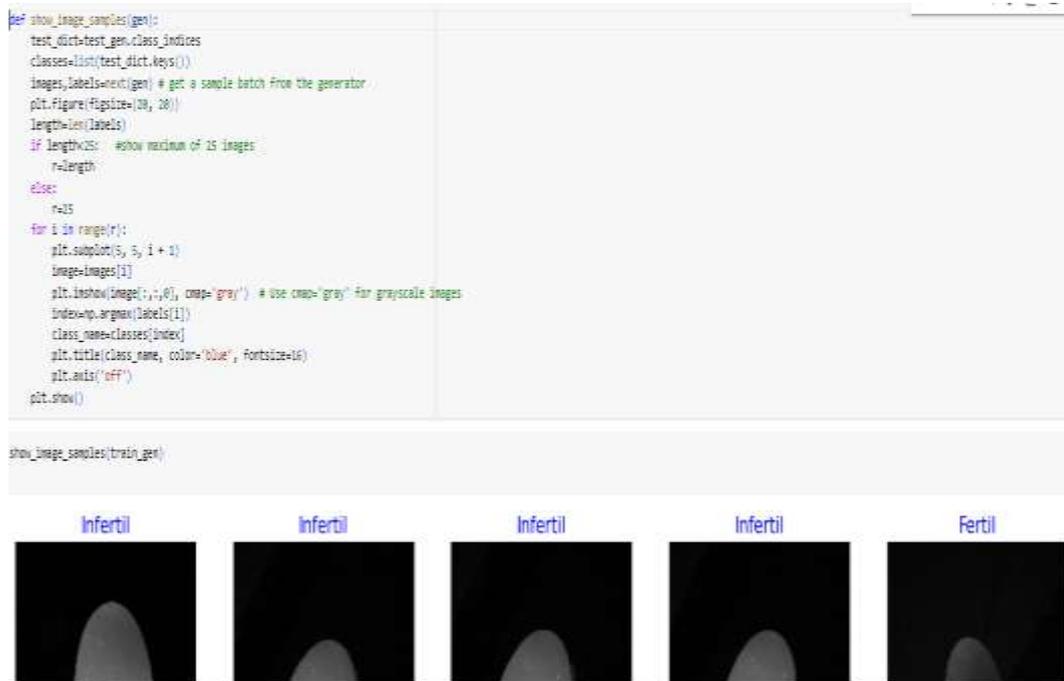
```

**Gambar 4.13 Data Generator**

1. Height dan Width berfungsi untuk menentukan tinggi dan lebar gambar yang akan diproses.
2. Channels berfungsi untuk menentukan jumlah saluran pada gambar. Karena menggunakan gambar grayscale, maka channels adalah 1.
3. Batch\_size berfungsi menentukan ukuran batch untuk pengolahan gambar dalam pelatihan model.
4. Img\_shape berfungsi untuk menentukan bentuk gambar, tinggi, lebar, dan channels menjadi satu tuple.
5. Img\_size berfungsi untuk menentukan ukuran gambar tujuan yang akan diubah selama preprocessing

#### 4.2.10. Visualisasi Data Generator

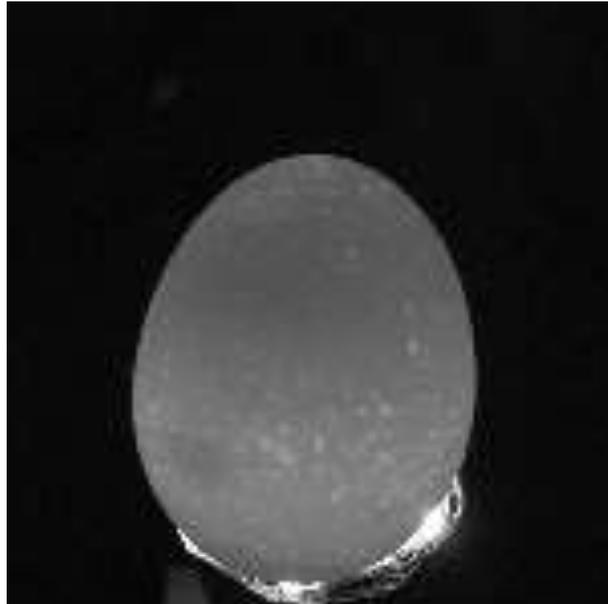
Tahap ini menjelaskan bagaimana visualisasi sampel gambar digunakan untuk memeriksa dan memastikan data yang dihasilkan oleh generator.



**Gambar 4.14 Visualisasi Data Generator**



**Gambar 4.15 Sebelum Greyscale**



**Gambar 4.16 Sesudah Greyscale**

#### 4.2.11. Model VGG16 Greyscale

Tahap ini menciptakan sebuah model jaringan saraf tiruan (CNN) untuk klasifikasi gambar dengan menggunakan TensorFlow dan Keras. Model dasar menggunakan VGG16, tetapi inputnya adalah gambar dalam format grayscale (satu kanal) dengan ukuran 150 x 150 piksel.

```
# Create model with grayscale input
IMG_SHAPE = (150, 150, 1) # Grayscale input shape

# Load base model VGG16
base_model = tf.keras.applications.VGG16(
    input_shape=(150, 150, 3), # VGG16 expects 3-channel input (RGB)
    include_top=False,
    weights='imagenet'
)

# Add a layer to convert grayscale to RGB
model = tf.keras.Sequential([
    tf.keras.layers.Conv2D(3, (3, 3), padding='same', input_shape=IMG_SHAPE), # Grayscale to RGB
    base_model,
    tf.keras.layers.Conv2D(32, 3, activation='relu'),
    tf.keras.layers.GlobalAveragePooling2D(),
    tf.keras.layers.Dense(2, activation='softmax')
])
```

**Gambar 4.17 Model VGG16 Greyscale**

Include\_top=false berfungsi untuk mengambil hanya bagian convolutional dari model VGG16 dengan menghapus lapisan penuh terhubung di atasnya yang memungkinkan untuk mengubah output model dengan sesuai. Weights=imagenet berfungsi untuk memanfaatkan transfer learning agar mempercepat dan meningkatkan pelatihan dengan menggunakan bobot yang telah dilatih pada dataset ImageNet.

#### 4.2.12. Konfigurasi Model Pelatihan

Tahap ini menjelaskan bagaimana model dikompilasi dengan menetapkan optimasi, fungsi loss, dan metrik yang sesuai untuk mencapai hasil pelatihan terbaik dalam klasifikasi gambar. Optimizer Adam adalah salah satu algoritme pengoptimalan yang paling banyak digunakan dalam pelatihan jaringan saraf karena memanfaatkan manfaat pengoptimalan AdaGrad dan RMSProp untuk memberikan kecepatan pembelajaran yang cepat dan stabil. Categorical crossentropy fungsi loss ini cocok untuk masalah klasifikasi gambar multi kelas. Sedangkan Accuracy digunakan untuk mengukur persentase prediksi yang benar.

```
model.compile(optimizer=tf.keras.optimizers.Adam(learning_rate=.001),
              loss='categorical_crossentropy',
              metrics=['accuracy'])
```

**Gambar 4.18 Konfigurasi Model**

#### 4.2.13. Training Model

Tahap ini menggunakan 10 epoch yang berarti bahwa model akan mengalami sepuluh kali pelatihan pada dataset. Setiap epoch, model diperbarui dan dioptimalkan menggunakan data yang telah dilatih, sementara data validasi

digunakan untuk mengevaluasi seberapa baik model dapat menggeneralisasi ke data baru.

```
history = model.fit(
    train_gen,
    validation_data = valid_gen,
    epochs = 10
)
```

Epoch 1/10  
/usr/local/lib/python3.10/dist-packages/keras/src/trainers/data\_adapters/py\_dataset\_adapter.py:122: UserWarning: Your self.\_warn\_if\_super\_not\_called()  
1/1 ----- 66s 66s/step - accuracy: 0.4250 - loss: 0.8323 - val\_accuracy: 0.4250 - val\_loss: 0.6964  
Epoch 2/10  
1/1 ----- 45s 45s/step - accuracy: 0.5750 - loss: 0.6479 - val\_accuracy: 0.4250 - val\_loss: 0.7063  
Epoch 3/10  
1/1 ----- 46s 46s/step - accuracy: 0.5750 - loss: 0.6299 - val\_accuracy: 0.4250 - val\_loss: 0.6624  
Epoch 4/10  
1/1 ----- 49s 49s/step - accuracy: 0.5750 - loss: 0.5873 - val\_accuracy: 0.6750 - val\_loss: 0.5812  
Epoch 5/10  
1/1 ----- 74s 74s/step - accuracy: 0.7500 - loss: 0.5238 - val\_accuracy: 0.9750 - val\_loss: 0.4844  
Epoch 6/10  
1/1 ----- 44s 44s/step - accuracy: 1.0000 - loss: 0.4534 - val\_accuracy: 1.0000 - val\_loss: 0.4035  
Epoch 7/10  
1/1 ----- 45s 45s/step - accuracy: 1.0000 - loss: 0.4005 - val\_accuracy: 1.0000 - val\_loss: 0.3427  
Epoch 8/10  
1/1 ----- 41s 41s/step - accuracy: 1.0000 - loss: 0.3566 - val\_accuracy: 1.0000 - val\_loss: 0.2936  
Epoch 9/10  
1/1 ----- 82s 82s/step - accuracy: 1.0000 - loss: 0.3162 - val\_accuracy: 1.0000 - val\_loss: 0.2502

**Gambar 4.19 Training Epoch**

#### 4.2.14. Evaluasi Model

Tahap ini hasil prediksi model dievaluasi, kesalahan diklasifikasikan dan ditampilkan, dan visualisasi seperti matriks kebingungan dan histogram kesalahan digunakan untuk menganalisis kinerja model secara lebih mendalam. `Class_dict` berfungsi mendapatkan informasi kelas dari generator data pengujian. `New_dict` berfungsi Membuat kamus yang memetakan angka kelas ke nama kelas untuk membuatnya lebih mudah dipahami.

```

def tr_plot(tr_data, start_epoch):
    #Plot the training and validation data
    tacc=tr_data.history['accuracy']
    tloss=tr_data.history['loss']
    vacc=tr_data.history['val_accuracy']
    vloss=tr_data.history['val_loss']
    Epoch_count=len(tacc)+ start_epoch
    Epochs=[]
    for i in range (start_epoch ,Epoch_count):
        Epochs.append(i+1)
    index_loss=np.argmin(vloss)# this is the epoch with the lowest validation loss
    val_lowest=vloss[index_loss]
    index_acc=np.argmax(vacc)
    acc_highest=vacc[index_acc]
    plt.style.use('fivethirtyeight')
    sc_label='best epoch= '+ str(index_loss+1 +start_epoch)
    vc_label='best epoch= '+ str(index_acc + 1+ start_epoch)
    fig,axes=plt.subplots(nrows=1, ncols=2, figsize=(8,8))
    axes[0].plot(Epochs,tloss, 'r', label='Training loss')
    axes[0].plot(Epochs,vloss, 'g', label='Validation loss' )
    axes[0].scatter(index_loss+1 +start_epoch,val_lowest, s=150, c= 'blue', label=sc_label)
    axes[0].set_title('Training and Validation Loss')
    axes[0].set_xlabel('Epochs')
    axes[0].set_ylabel('Loss')
    axes[0].legend()

```

### Gambar 4.20 Evaluasi Model

Proses pengumpulan, analisis, dan presentasi informasi terkait hasil evaluasi model akan dibahas di tahap ini. Metode ini termasuk pencetakan kesalahan, visualisasi distribusi kesalahan per kelas, matriks kebingungan, dan report klasifikasi.

```

def print_info( test_gen, preds, print_code, save_dir, subject ):
    class_dict=test_gen.class_indices
    labels= test_gen.labels
    file_names= test_gen filenames
    error_list=[]
    true_class=[]
    pred_class=[]
    prob_list=[]
    new_dict={}
    error_indices=[]
    y_pred=[]
    for key,value in class_dict.items():
        new_dict[value]=key # dictionary {integer of class number: string of class name}
    # store new_dict as a text fine in the save_dir
    classes=list(new_dict.values()) # list of string of class names
    dict_as_text=str(new_dict)
    dict_name= subject + '-' +str(len(classes)) +'.txt'
    dict_path=os.path.join(save_dir,dict_name)
    with open(dict_path, 'w') as x_file:
        x_file.write(dict_as_text)
    errors=0
    for i, p in enumerate(preds):
        pred_index=np.argmax(p)
        true_index=labels[i] # labels are integer values
        if pred_index != true_index: # a misclassification has occurred

```

**Gambar 4.21 Analisis Hasil Evaluasi Model dan Visualisasi**

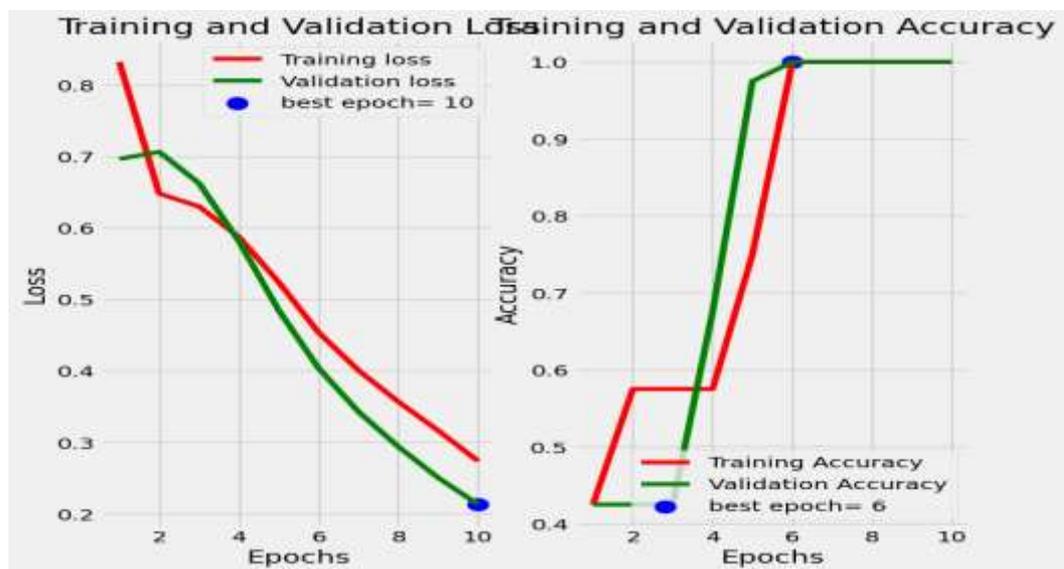
#### 4.2.15. Grafik Accuracy dan Loss

Tahap ini membahas metode yang digunakan untuk mengevaluasi model terhadap data pengujian, bagaimana hasil evaluasi dicetak, dan gambar yang digunakan untuk menunjukkan hasil pelatihan dan akurasi model. Berdasarkan data pelatihan yang telah dilakukan, fungsi `tr_plot` ini dapat digunakan untuk memvisualisasikan atau mengplot hasil pelatihan model. Jenis plot atau epoch yang akan ditampilkan dapat ditunjukkan oleh argumen `0`. `Save_dir` menetapkan direktori penyimpanan yang akan digunakan untuk menyimpan hasil atau file terkait lainnya. Nilai string `pest` disimpan oleh variabel `subjek`, yang dapat digunakan sebagai label atau identifikasi untuk model atau hasil yang dihasilkan. `Model.evaluate` mengevaluasi data pengujian (`test_gen`) dengan ukuran batch tertentu (`test_batch_size`) dan langkah-langkah pengujian (`test_steps`). Hasil evaluasi disimpan dalam variabel `acc` dan dikembalikan sebagai persentase akurasi.

Print\_in\_color mencetak hasil akurasi dalam format berwarna dan menyertakan persentase akurasi set pengujian dalam pesan.

```
tr_plot(history,0)
save_dir=r'./'
subject='pest'
acc=model.evaluate( test_gen, batch_size=test_batch_size, verbose=1, steps=test_steps, return_dict=False)[1]*100
msg=f'accuracy on the test set is {acc:5.2f} %'
print_in_color(msg, (0,255,0),(55,65,80))
## save_id=str(model_name + '-' + subject + '-' + str(acc)[:str(acc).rfind('.')+3] + '.hs')
# save_loc=os.path.join(save_dir, save_id)
# model.save(save_loc)
```

**Gambar 4.22 Pengecekan Akurasi Model pada Data Pengujian**

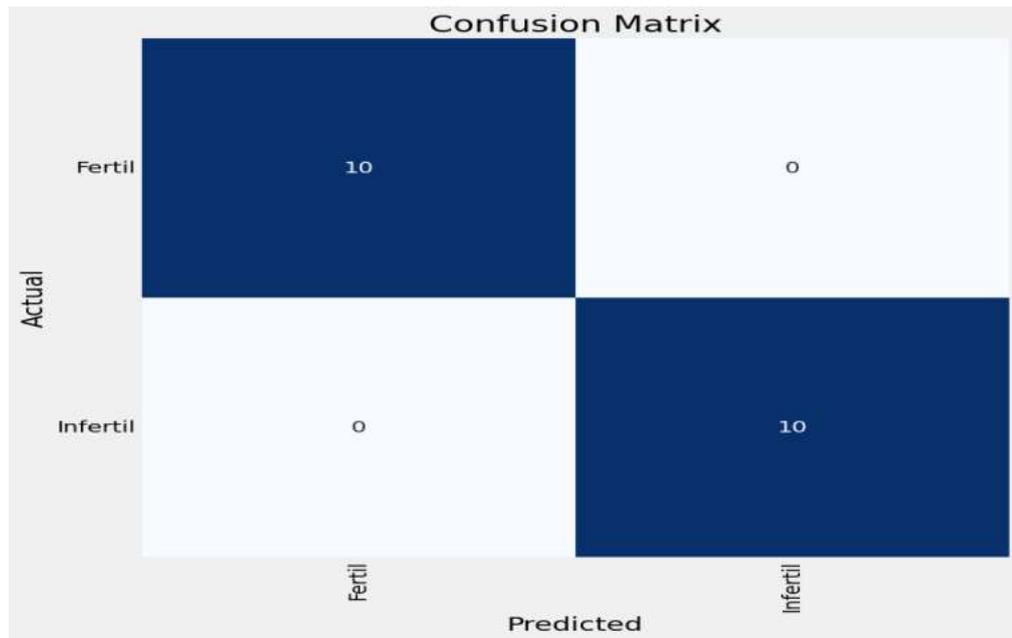


**Gambar 4.23 Grafik Akurasi dan loss**

Pada gambar diatas, grafik akurasi setiap epochnya semakin meningkat menandakan bahwa model sangat baik. Sedangkan grafik akurasi setiap epochnya semakin menurun menandakan bahwa kesalahan training semakin kecil.

#### 4.2.16. Testing Model

Pada tahapan ini dilakukan untuk mengetahui tingkat akurasi dan performa model VGG16.



**Gambar 4.24 Confusion Matriks**

Pada gambar diatas, confusion matrix menunjukkan bagaimana model klasifikasi berfungsi untuk membedakan telur fertil dan infertil. Matriks ini terdiri dari empat bagian, dan masing-masing menggabungkan prediksi model dan label yang sebenarnya. Pada matriks ini, dua sumbu yang disebut sebagai Fertil dan Infertil mewakili kategori aktual dan kategori prediksi. Sebagai hasil dari confusion matrix ini, dapat disimpulkan bahwa model klasifikasi yang dibuat dengan sangat baik, dengan tidak ada kesalahan klasifikasi. Semua sampel yang berhasil dikumpulkan diklasifikasikan dengan benar berdasarkan kategori aslinya. Ini menunjukkan bahwa model memiliki tingkat akurasi, ketepatan, recall, dan skor F1 yang sempurna (100%) untuk kedua kelas Fertil dan Infertil. Hasil ini menunjukkan bahwa model yang digunakan sangat efektif dan andal dalam membedakan telur yang fertil dan infertil berdasarkan data yang diberikan.

```

Classification Report:
-----

```

	precision	recall	f1-score	support
Fertil	1.00	1.00	1.00	10
Infertil	1.00	1.00	1.00	10
accuracy			1.00	20
macro avg	1.00	1.00	1.00	20
weighted avg	1.00	1.00	1.00	20

**Gambar 4.25 Report Classification**

Precision adalah rasio dari jumlah prediksi positif yang benar (True Positive + False Positive) terhadap total prediksi positif. Dalam report ini, precision untuk kelas Fertil adalah 1.00, yang menunjukkan bahwa semua prediksi model untuk kelas Fertil benar. Untuk kelas Infertil, precision juga adalah 1.00.

Recall adalah rasio dari jumlah prediksi positif yang benar (True Positive + False Negative) terhadap jumlah sampel yang benar-benar positif. Recall untuk kelas Fertil dan Infertil pada report ini adalah 1.00. Ini menunjukkan bahwa model dapat dengan benar mengidentifikasi semua sampel dari setiap kelas.

Dalam report ini, F1-Score untuk kedua kelas adalah 1.00, yang menunjukkan performa model yang sangat baik tanpa kesalahan prediksi. F1-Score adalah rata-rata harmonis dari precision dan recall, yang sangat berguna ketika ada ketidakseimbangan kelas.

Jumlah sampel nyata untuk setiap kelas adalah dukungan; dalam kasus ini, 10 sampel untuk kelas Fertil dan 10 sampel untuk kelas Infertil. Accuracy adalah rasio dari jumlah prediksi yang benar untuk kedua kelas Fertil dan Infertil terhadap jumlah sampel keseluruhan. Dalam model ini, akurasi adalah 1.00, yang berarti semua prediksi untuk kedua kelas benar.

Pada report ini, nilai macro average untuk precision, recall, dan F1-score adalah 1.00. Ini adalah rata-rata untuk semua kelas tanpa mempertimbangkan jumlah sampel yang ada di setiap kelas. Weighted average adalah rata-rata yang mempertimbangkan jumlah sampel di setiap kelas (support). Dalam report ini, weighted average memiliki nilai 1.00, yang menunjukkan bahwa hasil akhirnya tetap konsisten di semua metrik meskipun jumlah sampel mungkin tidak seimbang.

Secara keseluruhan, report ini menunjukkan bahwa model yang digunakan berhasil mengklasifikasikan kedua kelas dengan sangat baik, dengan tidak ada kesalahan prediksi pada dataset yang diuji.

#### 4.2.17. Save Model

Pada tahap ini dilakukan untuk menyimpan model yang telah dibuat yang nantinya model ini akan diupload untuk diimplementasikan pada website.

```
# Save the model to an .h5 file
model.save('model_telur.h5')

print("Model saved as model_telur.h5")

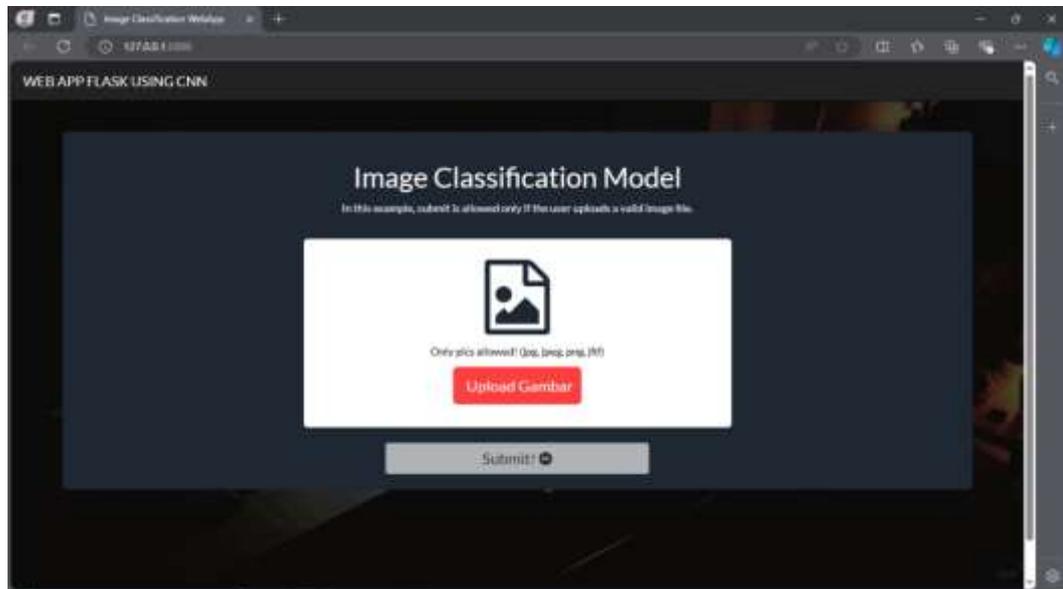
WARNING:absl:You are saving your model as
Model saved as model_telur.h5
```

**Gambar 4.26 Save Model**

### 4.3. Implementasi Website

Website ini dibangun menggunakan HTML untuk membuat antarmuka pengguna yang menarik dan interaktif serta menggunakan Flask memudahkan integrasi antara HTML dan logika aplikasi dan memungkinkan pengembangan fitur dinamis seperti pengolahan data dan penyampaian formulir.

### 4.3.1. Tampilan Halaman Utama (Index)



**Gambar 4.27 Halaman Utama**

Pada gambar diatas, merupakan halaman utama dari website. Pada halaman ini, untuk melakukan klasifikasi gambar, pengguna harus menginput atau upload gambar dan submit untuk memproses klasifikasi gambar.

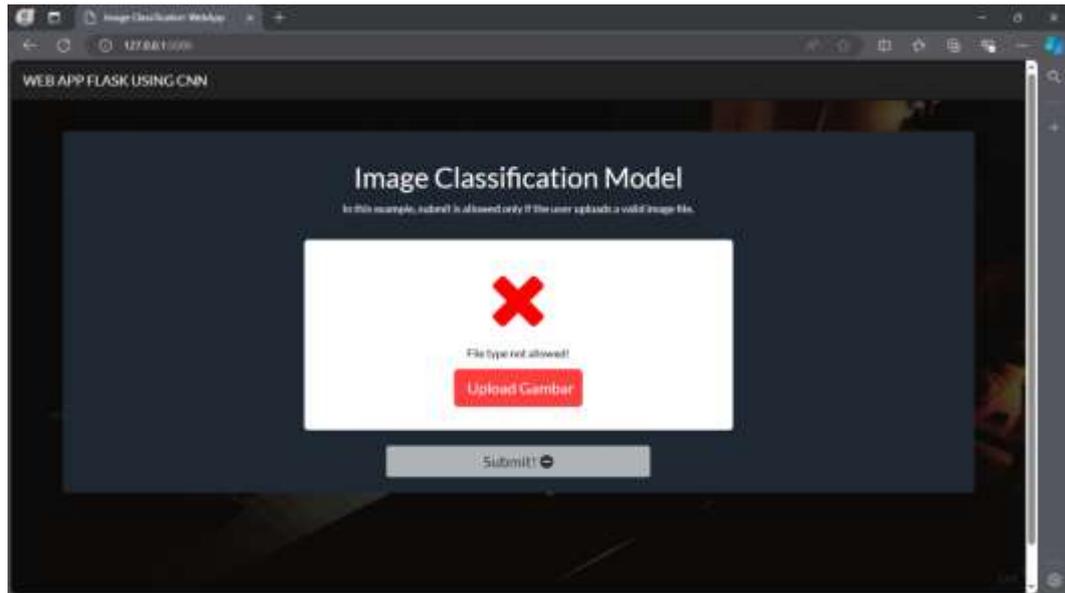
### 4.3.2. Tampilan Upload Sukses



**Gambar 4.28 Upload Sukses**

Pada gambar diatas, upload gambar harus berformat jpeg, jpg, png dan jfif. Apabila mengupload gambar yang tidak sesuai format, maka akan menampilkan icon error.

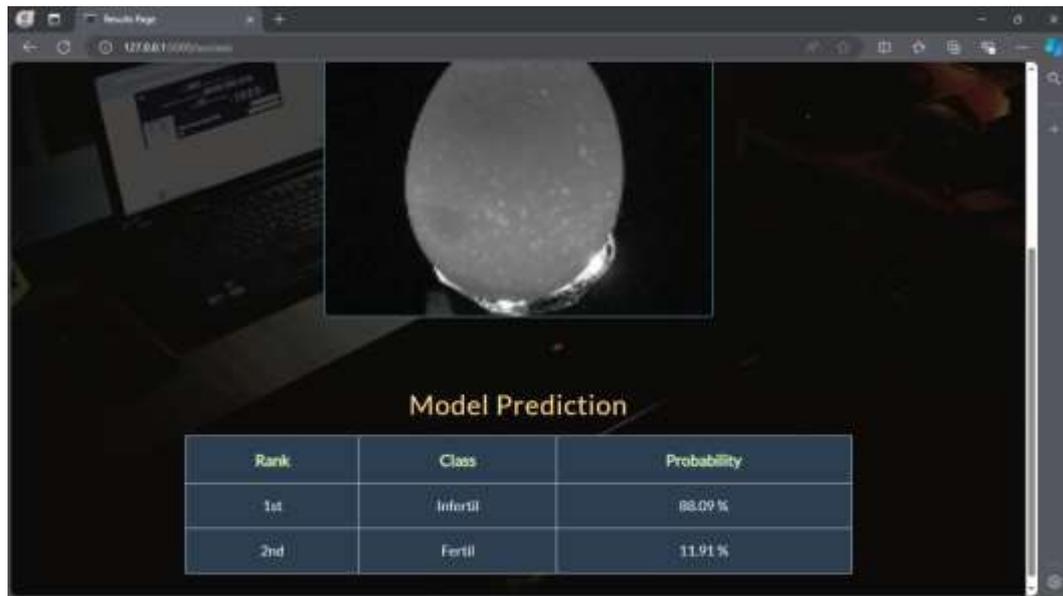
### 4.3.3. Tampilan Upload Gagal



**Gambar 4.29 Upload Gagal**

Pada gambar diatas, ketika pengguna upload gambar selain format jpeg, jpg, png dan jfif, maka akan menampilkan icon error.

#### 4.3.4. Tampilan Halaman Hasil (Sukses)



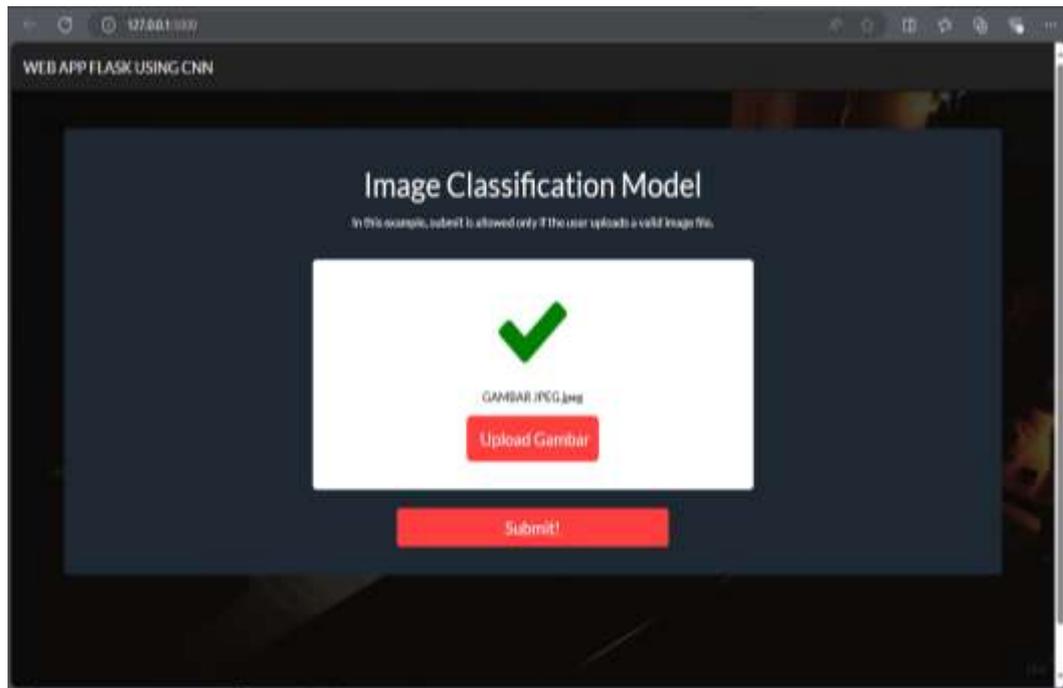
**Gambar 4.30 Hasil Klasifikasi**

Pada gambar diatas, menampilkan hasil klasifikasi dari gambar yang telah diupload.

#### 4.4. Pengujian Blackbox Testing

Pada tahap ini, peneliti melakukan pengujian menggunakan blackbox testing untuk mengecek semua fitur dapat berjalan dengan baik. Metode pengujian blackbox testing melakukan pengujian dari sudut pandang pengguna atau pihak eksternal tanpa melihat kode program atau struktur internalnya. Dengan metode ini, penguji hanya tahu input yang harus diberikan dan hasil yang diharapkan, tanpa mengetahui bagaimana proses tersebut bekerja di dalam sistem.

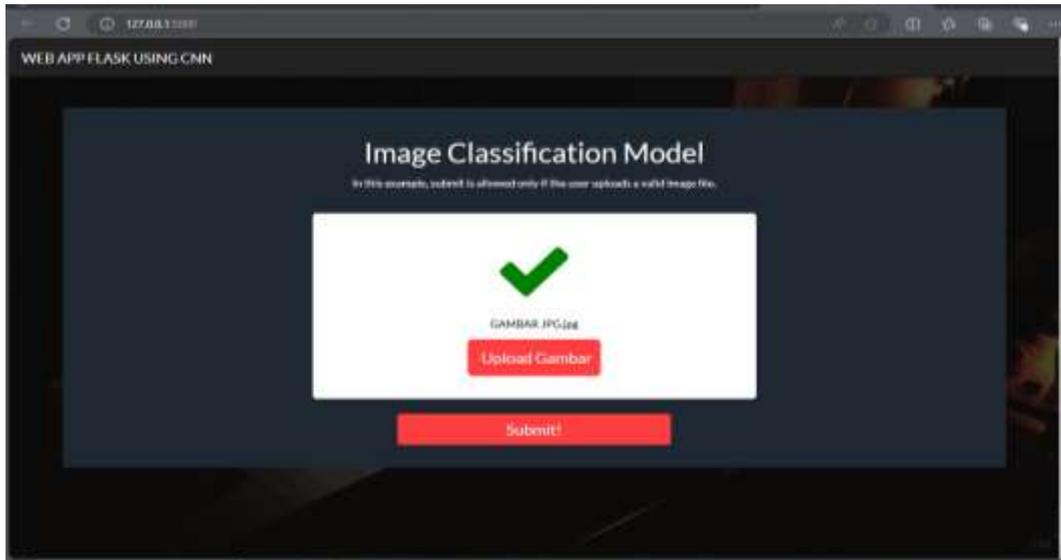
#### 4.4.1. Pengujian Upload JPEG



**Gambar 4.31 Pengujian JPEG**

Pengujian blackbox testing dilakukan dengan mengunggah gambar berformat JPEG. Setelah gambar terupload sesuai format JPEG, maka akan menampilkan icon sukses atau centang berwarna hijau.

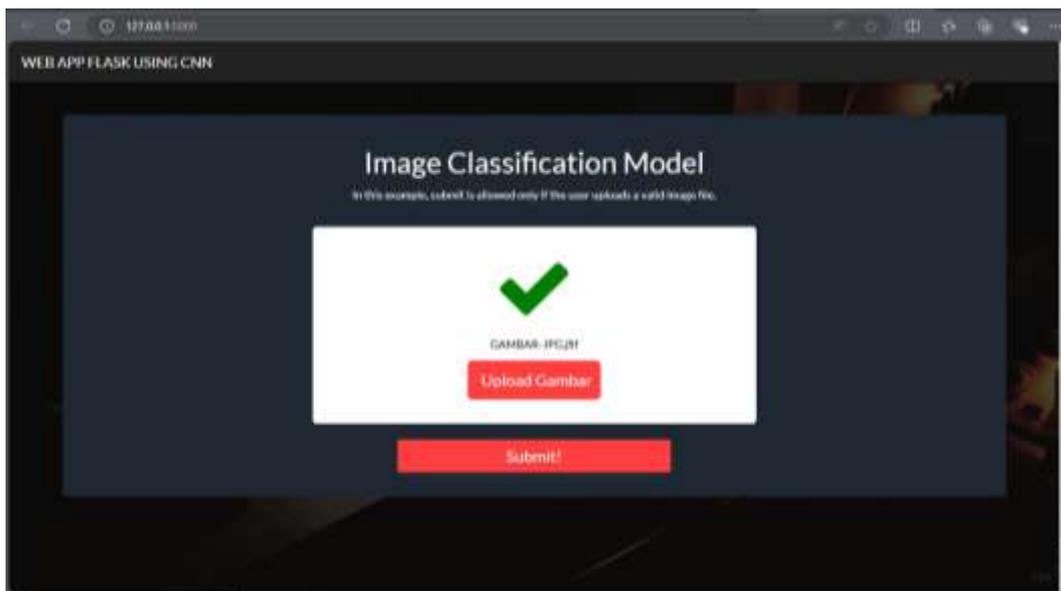
#### 4.4.2. Pengujian Upload JPG



**Gambar 4.32 Pengujian Jpg**

Pengujian blackbox testing dilakukan dengan mengunggah gambar berformat JPG. Setelah gambar terupload sesuai format JPG, maka akan menampilkan icon sukses atau centang berwarna hijau.

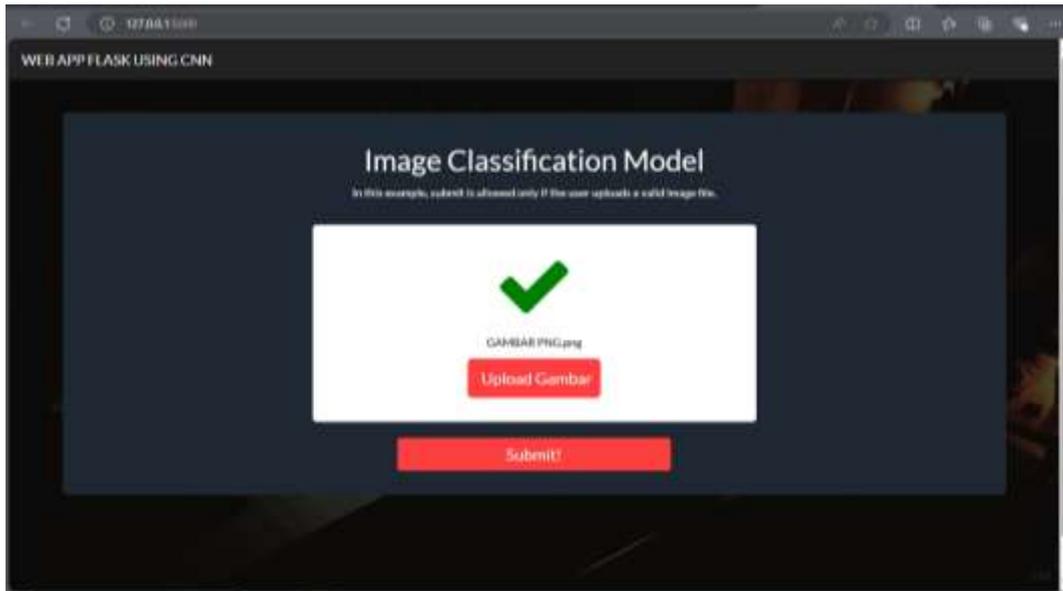
#### 4.4.3. Pengujian Upload JFIF



**Gambar 4.33 Pengujian Jfif**

Pengujian blackbox testing dilakukan dengan mengunggah gambar berformat JFIF. Setelah gambar terupload sesuai format JFIF, maka akan menampilkan icon sukses atau centang berwarna hijau.

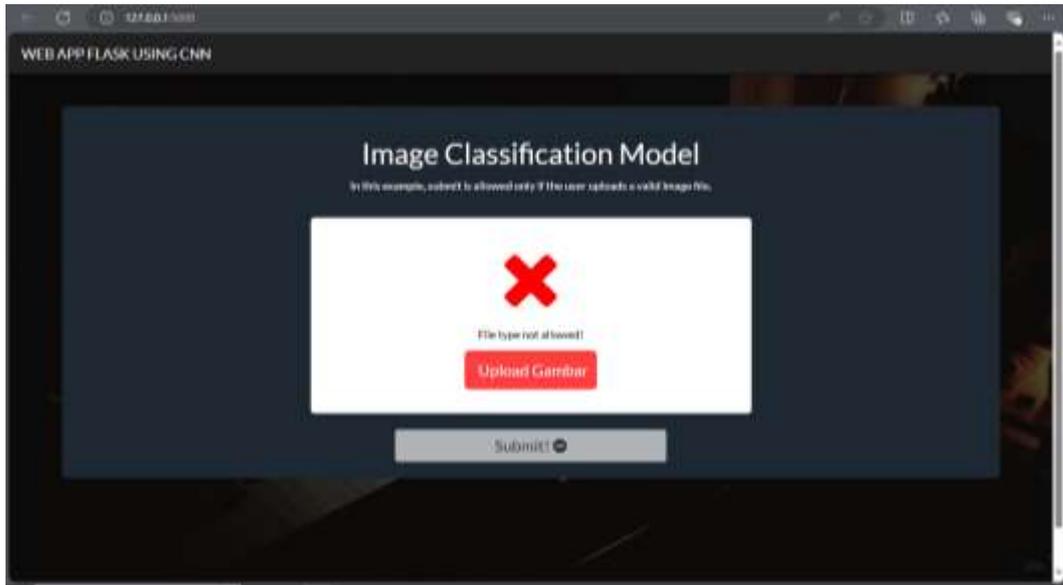
#### 4.4.4. Pengujian Upload PNG



**Gambar 4.34 Pengujian Png**

Pengujian blackbox testing dilakukan dengan mengunggah gambar berformat PNG. Setelah gambar terupload sesuai format PNG, maka akan menampilkan icon sukses atau centang berwarna hijau.

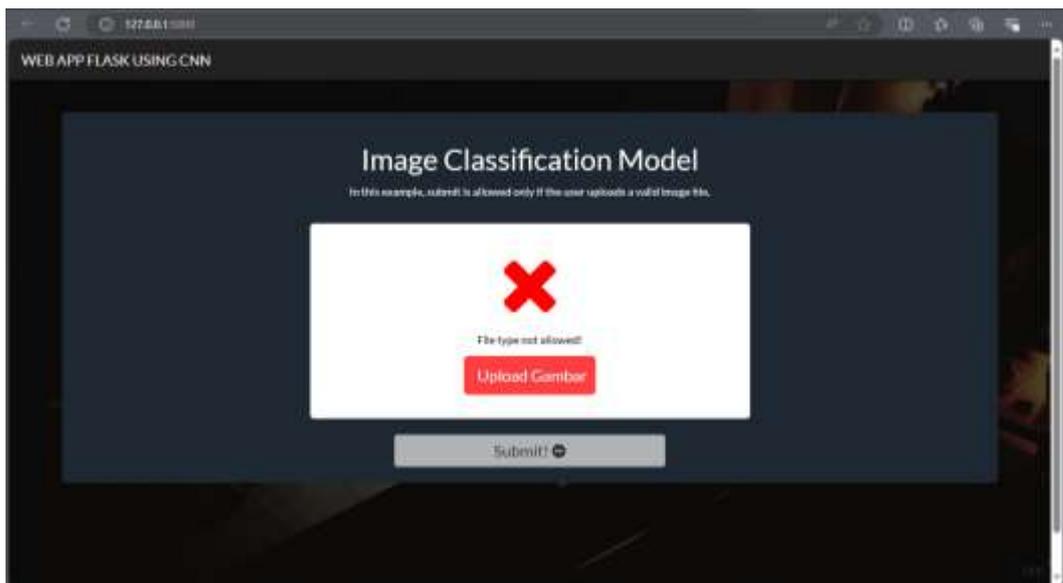
#### 4.4.5. Pengujian Upload DOC



**Gambar 4.35 Pengujian Doc**

Pengujian blackbox testing dilakukan dengan mengunggah file berformat DOC. Setelah file terupload, maka akan menampilkan icon gagal atau centang berwarna merah karena file tidak diizinkan.

#### 4.4.6. Pengujian Upload PDF



**Gambar 4.36 Pengujian Pdf**

Pengujian blackbox testing dilakukan dengan mengunggah file berformat PDF. Setelah file terupload, maka akan menampilkan icon gagal atau centang berwarna merah karena file tidak diizinkan.

#### 4.4.7. Pengujian Upload EXCEL



**Gambar 4.37 Pengujian Excel**

Pengujian blackbox testing dilakukan dengan mengunggah file berformat EXCEL. Setelah file terupload, maka akan menampilkan icon gagal atau centang berwarna merah karena file tidak diizinkan

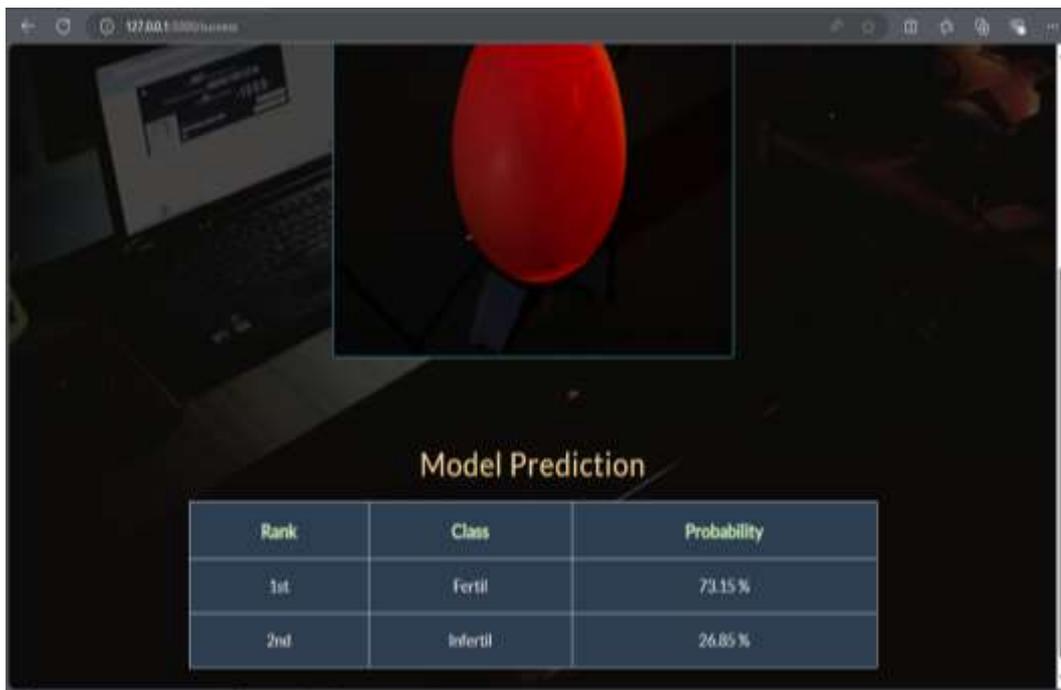
#### 4.4.8. Pengujian Upload MP4



**Gambar 4.38 Pengujian Mp4**

Pengujian blackbox testing dilakukan dengan mengunggah file berformat MP4. Setelah file terupload, maka akan menampilkan icon gagal atau centang berwarna merah karena file tidak diizinkan.

#### 4.4.9. Pengujian Klasifikasi Fertil

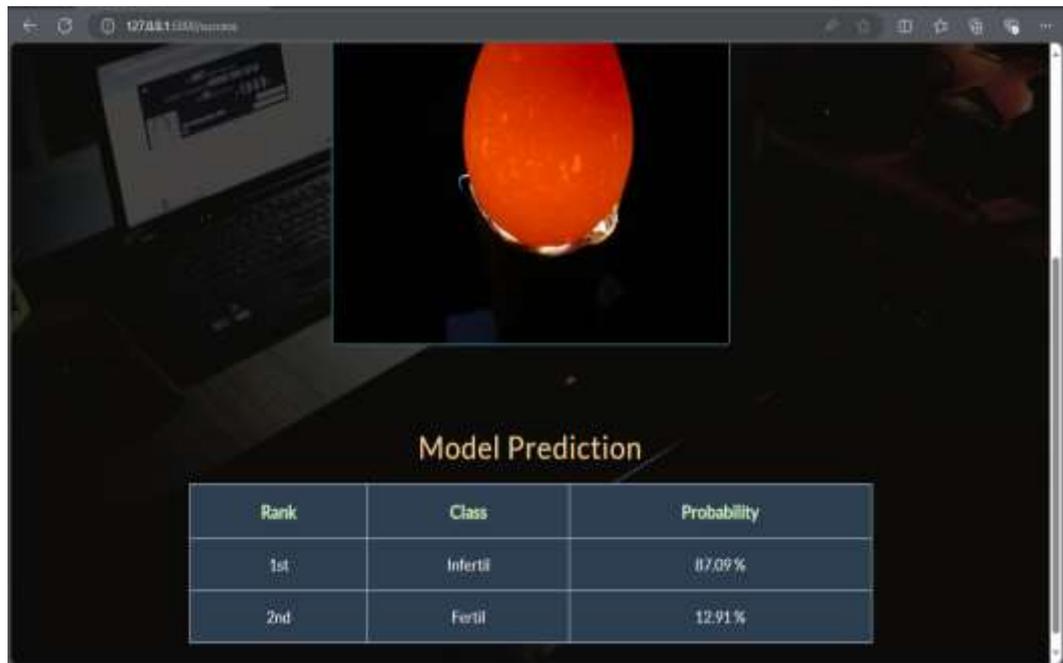


Rank	Class	Probability
1st	Fertil	73.15 %
2nd	Infertil	26.85 %

**Gambar 4.39 Pengujian Klasifikasi Fertil**

Pada pengujian blackbox testing, ketika mengunggah gambar telur fertil, sistem akan menampilkan hasil klasifikasinya pada halaman success.html. Pada pengujian ini, hasil klasifikasi menampilkan output 73.15% pada kelas fertil dan 26.85% pada kelas infertil. Sehingga output pada klasifikasinya yang sesuai adalah kelas fertil karena tingkat probabilitas yang tinggi.

#### 4.4.10. Pengujian Klasifikasi Infertil



**Gambar 4.40 Pengujian Klasifikasi Infertil**

Pada pengujian blackbox testing, ketika mengunggah gambar telur infertil, sistem akan menampilkan hasil klasifikasinya pada halaman success.html. Pada pengujian ini, hasil klasifikasi menampilkan output 87.09% pada kelas infertil dan 12.91% pada kelas fertil. Sehingga output pada klasifikasinya yang sesuai adalah kelas infertil karena tingkat probabilitas yang tinggi.

No	Skenario Pengujian	Uji Kasus	Hasil yang Diharapkan	Hasil Pengujian
1	Pengujian membuka halaman utama	Menjalankan flask dan membuka link flask	Menampilkan halaman utama	Sesuai Harapan
2	Mengunggah gambar dengan format JPEG	Mengunggah file gambar dengan format JPEG	Gambar berhasil diunggah dan	Sesuai Harapan

No	Skenario Pengujian	Uji Kasus	Hasil yang Diharapkan	Hasil Pengujian
			menampilkan icon sukses	
3	Mengunggah gambar dengan format JPG	Mengunggah file gambar dengan format JPG	Gambar berhasil diunggah dan menampilkan icon sukses	Sesuai Harapan
4	Mengunggah gambar dengan format PNG	Mengunggah file gambar dengan format PNG	Gambar berhasil diunggah dan menampilkan icon sukses	Sesuai Harapan
5	Mengunggah gambar dengan format JFIF	Mengunggah file gambar dengan format JFIF	Gambar berhasil diunggah dan menampilkan icon sukses	Sesuai Harapan
6	Mengunggah file dengan format DOC	Mengunggah file dengan format DOC	Gambar berhasil diunggah dan menampilkan icon gagal	Sesuai Harapan
7	Mengunggah file dengan format PDF	Mengunggah file dengan format PDF	Gambar berhasil diunggah dan menampilkan icon gagal	Sesuai Harapan
8	Mengunggah file dengan format EXCEL	Mengunggah file dengan	Gambar berhasil diunggah dan	Sesuai Harapan

No	Skenario Pengujian	Uji Kasus	Hasil yang Diharapkan	Hasil Pengujian
		format EXCEL	menampilkan icon gagal	
9	Mengunggah file dengan format MP4	Mengunggah file dengan format MP4	Gambar berhasil diunggah dan menampilkan icon gagal	Sesuai Harapan
10	Menekan tombol dashboard pada halaman success.html	Menekan tombol dashboard	Halaman Success.html redirect ke halaman index.html	Sesuai Harapan
11	Klasifikasi gambar telur fertil	Unggah gambar telur fertil	Menampilkan hasil klasifikasi dengan output fertil	Sesuai Harapan
12	Klasifikasi gambar telur infertil	Unggah gambar telur infertil	Menampilkan hasil klasifikasi dengan output infertil	Sesuai Harapan

**Tabel 4.2 Tabel Pengujian Blackbox Testing**

## **BAB V**

### **PENUTUP**

#### **5.1 Kesimpulan**

1. Algoritma CNN menggunakan model VGG16 yang dilatih pada gambar grayscale telur untuk mengklasifikasikan telur fertil dan infertil. Lapisan konvolusi dan pooling mengekstraksi fitur penting dari gambar. Hasil pelatihan berdasarkan confusion metriks seperti precision, recall, dan F1 Score menunjukkan probabilitas yang tinggi dalam klasifikasi telur fertil dan infertil. Hal ini tentunya menunjukkan kemampuan model dalam membedakan antara telur fertil dan infertil dengan sangat baik.
2. Merancang algoritma CNN melibatkan penggunaan arsitektur VGG16 untuk mengekstraksi fitur dari gambar grayscale telur. Dataset gambar telur fertil dan infertil digunakan untuk menguji model di Google Colab. Dengan menggunakan matrix confusion, evaluasi memastikan bahwa model dapat mengklasifikasikan telur dengan sangat akurat.

#### **5.2 Saran**

1. Peneliti menyarankan untuk peneliti selanjutnya dapat mengimplementasikan klasifikasi telur fertil dan infertil berbasis android.
2. Peneliti menyarankan untuk peneliti selanjutnya dapat menambahkan fitur realtime detection untuk klasifikasi gambar.
3. Peneliti menyarankan untuk peneliti selanjutnya dapat menggunakan beberapa algoritma lain untuk melakukan perbandingan algoritma.

## DAFTAR PUSTAKA

- Arnesia, P. D., Pratama, N. A., & Sjafrina, F. (2022). Aplikasi Artificial Intelligence Untuk Mendeteksi Objek Berbasis Web Menggunakan Library Tensorflow Js, React Js Dan Coco Dataset. *JSiI (Jurnal Sistem Informasi)*, 9(1), 62-69.
- Chicho, B. T., & Sallow, A. B. (2021). A comprehensive survey of deep learning models based on Keras framework. *Journal of Soft Computing and Data Mining*, 2(2), 49-62.
- Dirjen, S. K., Riset, P., Pengembangan, D., Dikti, R., Putu, I., & Pratama, A. E. (2017). Terakreditasi SINTA Peringkat 2 Pengujian Performansi Lima Back-End JavaScript Framework Menggunakan Metode GET dan POST. *Masa Berlaku Mulai*, 1(3), 1216–1225.
- Febrywinata, E. (2024). Pengenalan Dan Klasifikasi Jenis Buah Menggunakan Metode CNN Secara Sederhana Dengan Menggunakan Google Colab. *Merkurius: Jurnal Riset Sistem Informasi dan Teknik Informatika*, 2(4), 185-193.
- Firdaus, M. R. (2021). Penerapan Algoritma Convolutional Neural Network Dalam Klasifikasi Telur Ayam Fertil dan Infertil Berdasarkan Hasil Candling. *Jurnal Informatika Universitas Pamulang*, 5(4), 563. <https://doi.org/10.32493/informatika.v5i4.8556>
- Gazali, W., Soeparno, H., & Ohliati, ; Jenny. (n.d.). Penerapan Metode Konvolusi ..... (Wikaria Gazali; dkk) PENERAPAN METODE KONVOLUSI DALAM PENGOLAHAN CITRA DIGITAL.

- Indrawan, I. Gede, I. Made Sukada, and I. Ketut Suada. "Kualitas telur dan pengetahuan masyarakat tentang penanganan telur di tingkat rumah tangga." *Indonesia Medicus Veterinus* 1.5 (2012): 607-620.
- Nawawi, M. Z., Rahmat, R. F., Mohammad, D., & Syahputra, F. (2015). KLASIFIKASI TELUR FERTIL DAN INFERTIL MENGGUNAKAN JARINGAN SARAF TIRUAN MULTILAYER PERCEPTRON BERDASARKAN EKSTRAKSI FITUR WARNA DAN BENTUK FERTILE AND INFERTILE EGG CLASSIFICATION USING MULTI LAYER PERCEPTRON NEURAL NETWORK BASED ON COLOUR AND SHAPE FEATURE EXTRACTIONS (Vol. 4, Issue 2).
- Nihayatul Husna, I., Ulum, M., Kurniawan Saputro, A., Tri Laksono, D., & Neipa Purnamasari, D. (n.d.). Rancang Bangun Sistem Deteksi Dan Perhitungan Jumlah Orang Menggunakan Metode Convolutional Neural Network (CNN). *Seminar Nasional Fortei Regional*, 7.
- Pratiwi, A. O. C. (2023). Klasifikasi Jenis Anggur Berdasarkan Bentuk Daun Menggunakan Convolutional Neural Network Dan K-Nearest Neighbor. *Jurnal Ilmiah Teknik Informatika Dan Komunikasi*, 3(2), 201-224.
- Pakaja, F., Naba, A., & Purwanto, P. (2012). Peramalan Penjualan Mobil Menggunakan Jaringan Syaraf Tiruan dan Certainty Factor. *Jurnal EECCIS (Electrics, Electronics, Communications, Controls, Informatics, Systems)*, 6(1), 23-28.
- Rawat, A. (2020). A Review on Python Programming. *International Journal of Research in Engineering, Science and Management*, 3(12), 8-11.

- Shukla Shubhendu, S., & Vijay, J. (2013). Applicability of artificial intelligence in different fields of life. *International Journal of Scientific Engineering and Research*, 1(1), 28-35.
- Saifullah, Shoffan. "Analisis Perbandingan He Dan Clahe Pada Image Enhancement Dalam Proses Segmenasi Citra Untuk Deteksi Fertilitas Telur." *Jurnal Nasional Pendidikan Teknik Informatika: JANAPATI 9.1* (2020): 134-145.
- Wibawa, A. P., Guntur, M., Purnama, A., Fathony Akbar, M., & Dwiyanto, F. A. (2018). Metode-metode Klasifikasi. *Prosiding Seminar Ilmu Komputer Dan Teknologi Informasi*, 3(1).
- Wihandika, R. (2021). Deteksi masker wajah menggunakan metode adjacent evaluation local binary patterns. *Jurnal Resti (Rekayasa Sistem Dan Teknologi Informasi)*, 5(4), 705-712.
- Wijaya, A. A., & Prayudi, Y. (2010, June). Implementasi Visi Komputer Dan Segmentasi Citra Untuk Klasifikasi Bobot Telur Ayam Ras. In *Seminar Nasional Aplikasi Teknologi Informasi (SNATI)*.
- Zamachsari, F., & Puspitasari, N. (2021). Penerapan Deep Learning dalam Deteksi Penipuan Transaksi Keuangan Secara Elektronik. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, 5(2), 203–212. <https://doi.org/10.29207/resti.v5i2.2952>

