

Perancangan Sistem Deteksi Penyakit Kulit Pada Kucing Menggunakan Deep Learning Berbasis Android

Irgi Yoga Pangestu^{1*}, Shumaya Resty Ramadhani²

^{1,2} Program Studi Teknik Informatika, Politeknik Caltex Riau, Pekanbaru, Riau
Email: ^{1*}irgi19ti@mahasiswa.pcr.ac.id, ²shumaya@pcr.ac.id

(Naskah masuk: 23 Jul 2023, direvisi: 26 Sep 2023, diterima: 27 Sep 2023)

Abstrak

Penyakit kulit merupakan jenis penyakit yang sering dialami oleh makhluk hidup terutama kucing. Kucing salah satu hewan yang paling banyak dipelihara oleh manusia. Berdasarkan hasil survei berupa kuesoner yang disebar ke orang-orang yang memiliki kucing atau penyayang kucing, terdapat sekitar 65% dari 20 responden mengatakan kalau kucing peliharaannya pernah mengalami penyakit kulit dan sekitar 65% dari 20 responses mengatakan kesulitan mengenali penyakit kulit pada kucing. Pemilik atau pecinta kucing melakukan identifikasi penyakit kulit pada kucing dengan cara manual menggunakan indra penglihatan yaitu mata. Karena gejala penyakit kulit pada kucing memiliki kesamaan antar jenis penyakit, hal itu membuat pemilik atau pecinta kucing ragu atau salah mengenali jenis penyakit kulit pada kucing. Hal ini menyebabkan pecinta kucing salah memberikan pertolongan pertama atau pengobatan. Kondisi ini bisa membuat kucing bukan menjadi lebih sehat melainkan bisa bertambah parah. Solusi untuk mengatasi permasalahan tersebut untuk mengidentifikasi penyakit kulit pada kucing, penulis akan mengimplementasikan *Deep Learning* dengan menggunakan algoritma *Convolutional Neural Network* (CNN) untuk melakukan deteksi penyakit kulit pada kucing. Untuk mempermudah dalam pemakaiannya, dibuatlah dalam bentuk aplikasi android. Berdasarkan hasil uji *confusion matrix* didapatkan akurasi sebesar 78,33%, presisi sebesar 76,96%, dan *recall* sebesar 74,94%.

Kata Kunci: Android, *Convolutional Neural Network*, Deteksi, Penyakit Kulit, Kucing.

Design of Skin Disease Detection System in Cats Using Android-Based Deep Learning

Abstract

Skin disease is a type of disease that is often experienced by living things, especially cats. Cats are one of the most widely kept animals by humans. Based on the results of a survey in the form of a questionnaire distributed to people who have cats or cat lovers, there are about 65% of the 20 respondents said that their pet cats have experienced skin diseases and about 65% of the 20 responses said it was difficult to recognize skin diseases in cats. Cat owners or lovers identify skin diseases in cats manually using the sense of sight, namely the eyes. Because the symptoms of skin disease in cats have similarities between types of disease, it makes cat owners or lovers doubt or incorrectly recognize the type of skin disease in cats. This causes cat lovers to provide incorrect first aid or treatment. This condition can make the cat not get healthier but can get worse. The solution to overcome these problems to identify skin diseases in cats, the author will implement Deep Learning using the Convolutional Neural Network (CNN) algorithm to detect skin diseases in cats. To facilitate its use, it is made in the form of an android application. Based on the results of the confusion matrix test, the accuracy is 78.33%, the precision is 76.96%, and the recall is 74.94%.

Keywords: Android, Cats, *Convolutional Neural Network*, Detection, Skin Diseases.

I. PENDAHULUAN

Kucing merupakan salah satu hewan yang banyak dipelihara oleh manusia. Alasan manusia memelihara kucing adalah karena bentuk fisik yang lucu dan memiliki tingkah laku yang begitu menggemaskan [1]. Namun pemeliharaan kucing juga harus disertai dengan menjaga kesehatannya agar terhindar dari beragam jenis penyakit. Penyakit kulit adalah satu dari sekian banyak penyakit yang sering dialami oleh kucing. Penyakit kulit pada kucing dibagi menjadi 3 jenis yaitu tidak menular, menular sesama hewan, dan menular ke manusia. Bahkan ada penyakit kulit kucing yang sangat cepat menyebar pada manusia. Beberapa penyakit kulit pada kucing memiliki gejala yang hampir mirip seperti kucing yang akan menggaruk-garuk badannya dan rontoknya bulu pada bagian tertentu.

Menurut Undang-undang Republik Indonesia Nomor 18 Tahun 2009 Tentang Peternakan dan Kesehatan Hewan Pasal 1, Dokter hewan (Veteranian) adalah orang yang telah lulus dalam program pendidikan dokter hewan dari institusi pendidikan kedokteran hewan yang telah terakreditasi, sertifikat kompetensi, dan kewenangan medik veteriner dalam melaksanakan kesehatan hewan. Dokter hewan mayoritas membuka praktek di kota-kota saja. Hal ini membuat para pemilik atau pecinta kucing terlambat memberikan pertolongan pertama pada kucing kesayangannya. Dari hasil wawancara dengan dokter hewan Dimas Rizqo Sucitrawan yang berprofesi sebagai dokter hewan di *Animalia Vet Clinic* Pekanbaru, bahwa pasien kucing yang datang ke klinik rata-rata perhari ada 10 kucing.

Mendiagnosa penyakit kulit pada kucing sangat sulit bagi orang awam. Butuh bantuan tenaga ahli dibidangnya seperti dokter hewan untuk melakukan validasi kebenaran dari penyakit yang diderita oleh kucing. Berdasarkan hasil survei berupa kuesoner yang disebarakan ke orang-orang yang memiliki kucing atau penyayang kucing, terdapat sekitar 65% dari 20 responden mengatakan kalau kucing peliharaannya pernah mengalami penyakit kulit. Sekitar 65% dari 20 responden mengatakan kesulitan mengenali penyakit kulit pada kucing. Dan sekitar 90% dari 20 responden tertarik untuk menggunakan aplikasi pendeteksi penyakit kulit pada kucing. Dari hasil wawancara dengan dokter Dimas Rizqo Sucitrawan yang berprofesi sebagai dokter hewan di *Animalia Vet Clinic* Pekanbaru, bahwa penyakit kulit yang sering dialami oleh kucing adalah jamur, tungau, alergi, dan *scabies*.

Dari permasalahan yang ada pada latar belakang di atas, maka dilakukan pengembangan sistem deteksi penyakit kulit pada kucing menggunakan *deep learning* berbasis *android*. Penyakit kulit yang akan dideteksi ada empat jenis, yaitu infeksi jamur atau *ringworm*, *scabies*, jerawat kucing, dan tungau telinga. Data gambar penyakit kulit pada kucing didapatkan dari foto secara individu, *google image*, dan sosial media. *Deep learning* merupakan salah satu teknik dalam *machine learning* yang mengarahkan sebuah sistem komputer atau mesin untuk bekerja layaknya manusia pada umumnya. Dengan *deep learning* aplikasi ini akan mendeteksi penyakit kulit pada kucing seperti layaknya dokter hewan yang sedang mendeteksi penyakit kulit pada kucing. Metode *deep learning*

yang digunakan adalah *Convolutional Neural Network* (CNN) yang berguna untuk melakukan proses deteksi penyakit kulit. Karena pada penelitian ini menggunakan data gambar, maka metode CNN cocok untuk digunakan. Selain itu, dengan CNN pengguna tidak perlu melakukan *inputan* berupa gejala yang dialami oleh kucing. Cukup memasukkan gambar dari galeri penyimpanan *smartphone* atau tangkap gambar dengan kamera *smartphone*. Aplikasi ini akan dibangun berbasis android, guna untuk mempermudah dalam penggunaan. Dengan adanya aplikasi pendeteksi penyakit kulit pada kucing ini diharapkan dapat membantu para pecinta atau pemilik kucing dalam melakukan identifikasi penyakit kulit pada kucing, agar kucing dapat dengan mudah diobati.

II. TINJAUAN PUSTAKA

A. Penelitian Terdahulu

Penelitian mengenai deteksi penyakit kulit pada kucing atau penyakit kulit pada manusia atau yang mendekati sudah pernah dibuat sebelumnya dengan menggunakan metode dan algoritma yang berbeda. Penelitian pertama dengan judul *Diagnosa Penyakit Kulit Pada Kucing Menggunakan Metode Modified K-Nearest Neighbourhood* [2]. Penelitian ini bertujuan untuk mendiagnosa penyakit kulit pada kucing dengan menerima masukan berupa gejala-gejala yang dialami pada kucing, yang nantinya masukan tersebut akan diolah oleh sistem dengan algoritma *Modified K-Nearest Neighbourhood* (K-NN) dengan keluaran berupa hasil diagnosa penyakit kulit pada kucing yang diderita. Dalam pengujian pakar menggunakan 120 data uji dengan data latih (*epoch*) dibagi menjadi 3, yaitu 20, 30, dan 40. Metode ini menghasilkan nilai akurasi 89% hingga 99%.

Penelitian kedua dengan judul *Sistem Pakar Diagnosa Penyakit Kulit Pada Kucing Menggunakan Metode Naive Bayes Berbasis Web* [3]. Pada penelitian ini sistem pakar menggunakan metode *Naive Bayes* yang akan mendiagnosa penyakit kulit pada kucing yang berasal dari konsultasi user ke sistem. Ketika dilakukan pengujian akurasi diagnosa penyakit didapatkan tingkat akurasi sebesar 80%.

Penelitian ketiga dengan judul *Deteksi Penyakit Kulit Wajah Menggunakan Tensorflow Dengan Metode Convolutional Neural Network* [4]. Penelitian ini menggunakan algoritma *Convolutional Neural Network* (CNN) untuk melakukan deteksi penyakit kulit pada wajah. *Dataset* yang digunakan sebanyak 700 citra penyakit kulit wajah dengan jumlah 20 jenis penyakit dan diperoleh ketepatan tertinggi 99,91% dengan persentase rata-rata ketepatan 80%.

Penelitian keempat dengan judul *Sistem Klasifikasi Penyakit Kulit Menggunakan Convolutional Neural Network* (CNN) [5]. Penelitian ini menggunakan metode *Convolutional Neural Network* (CNN) untuk mengklasifikasi penyakit kulit. Ada 1500 data citra yang terdiri dari 1125 data latih dan 375 data uji yang digunakan dalam penelitian ini. Hasil performansi sistem terbaik pada penelitian ini berdasarkan keseluruhan skenario pengujian parameter didapatkan saat *hidden layer* berjumlah 5, menggunakan *adam optimizer*, dan

nilai *learning rate* sebesar 0,001 dengan sistem dapat mencapai akurasi sebesar 96,53%, *precision* 95%, *recall* 95%, *f1-score* 95% dan *loss* hanya 0,2486.

B. Penyakit Kulit Kucing

Penyakit kulit merupakan satu dari banyaknya jenis penyakit yang dialami oleh kucing. Penyakit kulit sangat rentan menular kepada kucing lain dan bahkan manusia. Beberapa gejala ketika kucing mengalami penyakit kulit dapat dilihat dari perubahan pada kucing seperti kebotakan, kulit kemerahan, luka pada kulit, dan berbau tidak sedap [6]. Walaupun penyakit kulit pada kucing ada beragam, tetapi pada proyek akhir ini hanya 4 jenis penyakit kulit yang digunakan, antara lain yaitu:

- 1) Infeksi Jamur atau *Ringworm*
- 2) *Scabies*
- 3) Jerawat Kucing
- 4) Tungau Telinga

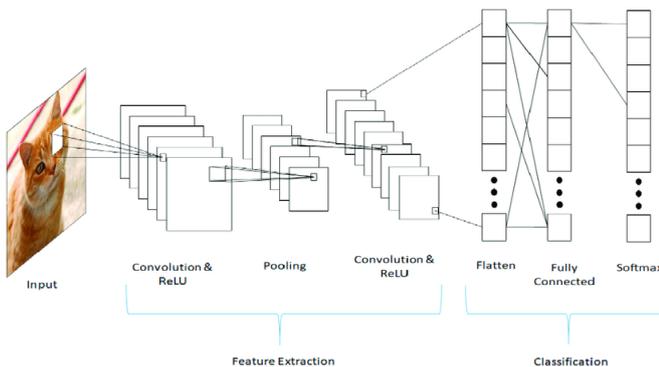
C. Deep Learning

Deep Learning merupakan subbidang dari *machine learning* yang metode algoritmanya terinspirasi dari struktur otak manusia yang neuron-neuronnya terkoneksi satu sama yang lain sehingga membentuk jaringan neuron yang sangat rumit. *Deep learning* memanfaatkan algoritma *artificial neural network* (ANN) yang terinspirasi dari jaringan biologis otak manusia [7]. Ada beberapa jenis jaringan saraf tiruan yang termasuk kedalam *deep learning*, yaitu:

- 1) *Convolutional Neural Network* (CNN)
- 2) *Recurrent Neural Network* (RNN)
- 3) *Long Short Term Memory Network* (LSTM)
- 4) *Self Organizing Maps* (SOM)

D. Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) merupakan jenis algoritma *deep learning* yang lebih banyak digunakan pada data *image* atau citra. CNN didesain untuk mengolah data dua dimensi. CNN adalah pengembangan dari *Multilayer Perception* (MLP) yang termasuk kedalam *neural network* bertipe *feed forward* (tidak berulang) [8]. CNN memiliki beberapa *layer* yang difungsikan untuk melakukan filter yang disebut proses *training*. Pada proses *training* terdapat 3 tahapan yang harus dilakukan yaitu *Convolutional Layer*, *Pooling layer*, dan *Fully connected layer* [9].



Gambar 1. Contoh Arsitektur CNN

Dari Gambar 1 di atas arsitektur CNN terdiri dari beberapa lapisan dengan fungsi yang berbeda-beda. Tetapi, secara umum CNN terbagi atas 2 bagian, yaitu *Feature Extraction* atau dan *Classification*.

a) *Feature Extraction*

Proses yang terjadi dibagian *feature extraction* atau sering disebut juga dengan *feature learning* adalah bagian untuk mengubah dari sebuah gambar atau citra menjadi *feature* dalam bentuk angka-angka. Pada *Feature extraction* memiliki beberapa proses yang terdiri dari:

1) *Convolutional Layer*

Convolutional layer merupakan proses utama dalam algoritma *Convolutional Neural Network* (CNN) yang bertugas untuk melakukan operasi konvolusi pada output dari *layer* sebelumnya [10]. *Convolution layer* dijalankan pada data *input* menggunakan sebuah filter atau kernel yang kemudian digunakan untuk memetakan fitur. Misalnya sebuah *image* atau gambar masukkan memiliki resolusi pixel 32x32 terdapat 3 saluran warna, yaitu *Red*, *Green*, dan *Blue* (RGB), maka akan terdapat 32x32x3 *array*. Hal ini dikarenakan CNN memiliki lapisan *neuron* yang terdiri dari 3 dimensi, yaitu lebar, tinggi, dan kedalaman. Operasi *convolution layer* dilakukan dengan menggeser filter dan operasi "dot" antara data masukan dan nilai dari filter. Pada Gambar 2 di bawah merupakan contoh operasi perhitungan pada *convolutional layer*. Untuk sebelah kiri merupakan *layer* awal dan sebelah kanan adalah hasil *layer* setelah dilakukan operasi filter.

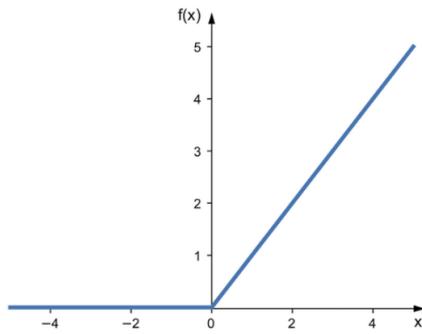
1	1	1	0	0
0x1	1x0	1x1	1	0
0x0	0x1	1x0	1	1
0x1	0x0	1x1	1	0
0	1	1	0	0

4	3	4
2		

Gambar 2. Contoh Operasi Convolution Layer

2) *Activation ReLU*

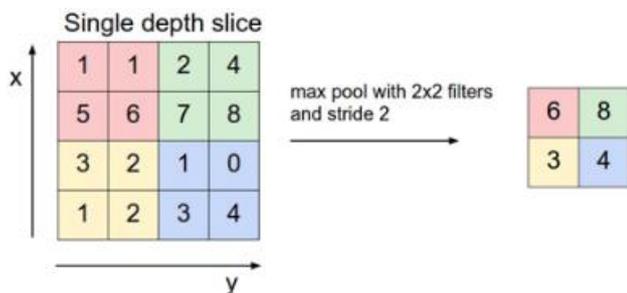
Activation ReLU (*Rectified Linear Unit*) merupakan lapisan aktivasi pada CNN yang mengaplikasikan fungsi $f(x) = \max(0, x)$ untuk melakukan *Thresholding* dengan nilai 0 terhadap nilai piksel pada *input* citra [11]. *Activation ReLU* berfungsi menghitung aktivasi dengan membatasi *input* pada nol. Unit linear yang diperbaiki adalah keluaran 0, jika masukan kurang dari 0 maka aktivasi *ReLU* akan mengubah piksel menjadi 0 guna untuk meningkatkan kualitas untuk meminimalisir kesalahan yang akan terjadi. Dari Gambar 3 di bawah dapat dilihat garis biru yang menggambarkan apabila nilai negatif dimasukkan akan menghasilkan nilai 0 dan apabila masukkan bernilai positif akan menghasilkan nilai yang sama dengan nilai masukkan tersebut.



Gambar 3. ReLu Activation Function

3) Pooling Layer

Fungsi dari proses Pooling layer adalah untuk mengurangi dimensi secara terus menerus serta mengurangi jumlah parameter dan komputasi dalam jaringan. Hal ini akan berdampak pada waktu training yang menjadi lebih singkat dan mengontrol terjadinya overfitting. Penggunaan pooling layer bertujuan mereduksi ukuran citra [12]. Ukuran citra ditentukan dari jumlah stride. Jenis pooling layer yang sering digunakan pada algoritma CNN adalah max-pooling. Sebagai contoh menggunakan max pooling 2x2 dengan stride 2, maka setiap pergeseran filter nilai yang diambil adalah nilai terbesar pada area 2x2.



Gambar 4. Ilustrasi Pada Max-Pooling Layer

Dari ilustrasi pada Gambar 4 di atas dapat dilihat pada layer sebelah kiri dilakukan pembagian 4 area dengan masing-masing ukuran 2x2 untuk setiap area. Dan pada setiap area akan diambil nilai terbesar dan dijadikan layer baru (layer kanan).

b) Classification

Pada bagian kedua adalah classification yang berisi proses mengklasifikasi pada setiap neuron yang telah dilakukan ekstraksi pada tahap feature extraction. Bagian ini terdiri dari:

1) Flatten

Tahapan flatten adalah tahapan yang berguna untuk merubah matriks dari pooling layer menjadi satu kolom saja (vektor tunggal). Vektor ini akan menjadi bagian dari input layer ke fully-connected layer.

2) Fully-Connected Layer

Feature map dari proses sebelumnya masih berbentuk multidimensional array, sehingga harus dilakukan proses

“flatten” atau reshape feature map menjadi sebuah vector agar bisa digunakan sebagai input dari fully-connected layer [7]. Fully-Connected Layer merupakan layer semua neuron aktivasi dari layer sebelumnya terhubung semua dengan neuron di lapisan selanjutnya yang membentuk jaringan tiruan. Setiap nilai berkontribusi untuk memprediksi seberapa kuat suatu nilai cocok dengan kelas tertentu.

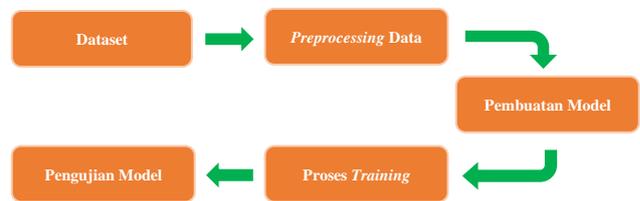
3) Dropout

Dropout adalah salah satu cara untuk mencegah terjadinya overfitting dan mempercepat proses learning [8]. Overfitting merupakan kondisi yang mengacu pada masalah saat model dilatih (proses training) yang mendapatkan persentase baik, tetapi terjadi ketidaksesuaian pada proses prediksi.

III. METODE PENELITIAN

A. Alur Deep Learning

Alur deep learning merupakan gambaran urutan proses deep learning yang akan digunakan pada penelitian ini. Proses ini harus dilakukan secara berurutan dan tidak boleh ada yang terlewatkan. Dari Gambar 5 di bawah dapat dilihat gambaran alur deep learning yang akan digunakan pada penelitian ini.



Gambar 5. Alur Deep Learning

a) Dataset

Dataset merupakan data yang dikumpulkan dari berbagai sumber seperti sosial media, google image, dan foto secara langsung. Dataset digunakan untuk membuat model. Data yang dikumpulkan berupa gambar atau citra penyakit kulit pada kucing dengan total 400 data.

b) Preprocessing Data

Preprocessing data merupakan alur yang berguna untuk mengubah ukuran dan dimensi citra dari dataset yang sudah dikumpulkan menjadi sama. Perubahan tersebut dilakukan karena citra masih memiliki ukuran dan dimensi yang berbeda-beda.

c) Pembuatan Model

Pembuatan model adalah tahap terpenting sebelum melakukan training. Model ini dibuat dari berbagai layer, seperti convolution layer, pooling layer, flatten layer, dense layer, dan dropout layer.

d) Proses Training

Proses training adalah proses untuk mempelajari model yang sebelumnya sudah dibuat. Jumlah proses training dilakukan berdasarkan jumlah epoch.

e) Pengujian Model

Pengujian model merupakan alur terakhir setelah model sudah berhasil di-*training*. Pengujian model berguna untuk melihat seberapa besar akurasi model yang sudah dibuat dapat mendeteksi penyakit kulit pada kucing.

B. Metode Pengujian

a) *Confusion matrix*

Confusion matrix merupakan tahap pengujian akurasi yang dilakukan untuk menghitung nilai akurasi atau kecocokan dari data baru yang masuk kedalam aplikasi dengan data yang sudah dilatih sebelumnya [13]. Untuk bentuk *confusion matrix* secara umum dapat dilihat pada Tabel 1 di bawah ini.

Tabel 1. Bentuk *Confusion Matrix*

		Nilai Sebenarnya	
		Positif	Negatif
Nilai Prediksi	Positif	True Positive (TP)	False Positive (FP)
	Negatif	False Negative (FN)	True Negatif (TN)

Penjelasan dari tabel 1 diatas adalah:

- True Positive* (TP) merupakan jumlah prediksi positif yang benar
- True Negative* (TN) merupakan jumlah prediksi negatif yang benar
- False Positive* (FP) merupakan jumlah prediksi positif yang salah
- False Negative* (FN) merupakan jumlah prediksi negatif yang salah

Rumus:

$$\text{Akurasi} = \frac{\text{Total TP}}{\text{Total Data}} \times 100\% \tag{1}$$

$$\text{Presisi} = \frac{\text{TP}}{\text{TP} + \text{FP}} \times 100\% \tag{2}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \times 100\% \tag{3}$$

b) *Black Box Testing*

Black box testing merupakan pengujian kualitas perangkat lunak yang terfokus kepada fungsionalitas suatu perangkat lunak [14]. Teknik *black box testing* memungkinkan untuk memperoleh serangkaian masukan dari pengguna yang dimana dalam *black box testing* berusaha menemukan kesalahan dari komponen yang terkandung dalam perangkat lunak.

Black box testing cenderung digunakan untuk menemukan beberapa hal seperti berikut:

- Fungsi yang tidak benar
- Kesalahan antarmuka (*interface errors*)
- Kesalahan pada struktu data
- Kesalahan performansi (*performance errors*)
- Kesalahan inialisasi dan terminasi

c) Uji Analisis

Uji analisis dilakukan untuk menentukan pengaruh dari beberapa kondisi terhadap hasil akurasi klasifikasi pada aplikasi. Perbandingan ini dilakukan dengan tujuan untuk mendapatkan kesimpulan dari seluruh pengujian kondisi yang dilakukan. Dari kesimpulan itu akan mendapatkan akurasi yang tinggi dari beberapa kondisi pada Tabel 2, mulai dari segi spesifikasi *smartphone*, kualitas kamera, dan jarak tangkap gambar. Selain itu pencahayaan juga akan diuji untuk melihat pengaruh dari hasil akurasi yang akan didapatkan.

Tabel 2. Kondisi Pada Pengujian Analisis

No	Spesifikasi Smartphone	Kualitas Kamera	Jarak Tangkap Gambar
1.	Xiaomi Redmi 9T, RAM 4 GB, Android 12, CPU Octa-core Max2.0GHz	42 MP	<ul style="list-style-type: none"> • 5 cm • 10 cm • 20 cm
2.	Xiaomi Redmi 10C, RAM 4GB, Android 11, CPU Octa-core	50 MP	<ul style="list-style-type: none"> • 5 cm • 10 cm • 20 cm
3.	OPPO A52, RAM 4GB, Android 10, CPU Octa-core	12 MP	<ul style="list-style-type: none"> • 5 cm • 10 cm • 20 cm

IV. HASIL DAN PEMBAHASAN

A. Dataset

Pada penelitian ini, *dataset* yang digunakan berupa kumpulan gambar penyakit kulit pada kucing yang diambil dengan beberapa sumber, yaitu cara manual menggunakan kamera *smartphone*, dari *google image*, dan dari sosial media (*instagram*). Gambar 6 di bawah merupakan contoh gambar kucing sakit yang digunakan sebagai *dataset*.





Gambar 6. Contoh Gambar Kucing Sakit

Dataset yang dikumpulkan berjumlah 400 gambar dengan masing-masing jenis penyakit 100 gambar. Pengambilan gambar harus dilakukan dari beberapa arah seperti depan, atas, kiri, dan kanan. Dengan pengambilan gambar dari arah yang berbeda akan meningkatkan akurasi ketika pendeteksian penyakit. *Dataset* akan dibagi menjadi 2 bagian, yaitu data *training* (latih) dan data *testing* (uji) dengan bobot 85% data *training* dan 15% data *testing*. Pada Tabel 3 di bawah menunjukkan pengambilan gambar penyakit kulit pada kucing yang digunakan sebagai *dataset*.

Tabel 3. Daftar Jenis Penyakit Kulit Pada Kucing

No	Jenis Penyakit	Data Training	Data Testing
1	Infeksi Jamur atau Ringworm	85	15
2	Scabies	85	15
3	Jerawat Kucing	85	15
4	Tungau Telinga	85	15
Jumlah		340	60

A. Preprocessing Data

Citra yang sudah di *input*, selanjutnya akan di-*resize*. *Resize* berguna untuk mengubah agar dimensi pada citra memiliki ukuran, tinggi, dan rasio yang sama. Pada *Dataset* yang sudah dikumpulkan memiliki variasi *size* yang berbeda-beda. Karena, seluruh *dataset* berasal dari gambar-gambar yang di-*crop*. *Resize* ini dilakukan karena dapat mempengaruhi dalam waktu proses *training* dan hasil akurasi.

Tabel 4. Percobaan Ukuran *Resize*

Ukuran <i>Resize</i>	Waktu <i>Trining</i>	Akurasi
256x256	3 Menit 54 detik	0,63
180x180	2 Menit 27 detik	0,63
160x160	1 Menit 30 detik	0,78

Tabel 4 di atas menggambarkan beberapa ukuran *resize* yang dicoba. Dapat dilihat bahwa ukuran *resize* sangat mempengaruhi waktu *training* dan akurasi. Dari hasil percobaan tersebut, diambil ukuran *resize* yang digunakan dalam pembuatan model adalah 160 x160 *pixel*.

B. Pembuatan Model

Model CNN yang dibangun memiliki beberapa *layer*, yaitu *convolution layer*, *pooling layer*, *flatten layer*, *dense layer*,

dan *dropout layer*. Pada Gambar 7 di bawah merupakan model yang akan digunakan dalam penelitian ini yang didapatkan dari beberapa kali percobaan model dengan mencari hasil akurasi terbaik.

```

model = Sequential()
model.add(Conv2D(16, (3,3), input_shape= (160,160,3)))
model.add(Activation("relu"))
model.add(MaxPooling2D((2,2)))
model.add(Dropout(0.1))

model.add(Conv2D(32, (3,3),))
model.add(Activation("relu"))
model.add(MaxPooling2D((2,2)))
model.add(Dropout(0.1))

model.add(Conv2D(64, (3,3),))
model.add(Activation("relu"))
model.add(MaxPooling2D((2,2)))
model.add(Dropout(0.1))

model.add(Conv2D(128, (3,3),))
model.add(Activation("relu"))
model.add(MaxPooling2D((2,2)))
model.add(Dropout(0.1))

model.add(Flatten())
model.add(Dense(512))
model.add(Activation("relu"))
model.add(Dropout(0.5))
model.add(Dense(number_of_class))
model.add(Activation("softmax"))
    
```

Gambar 7. Model CNN

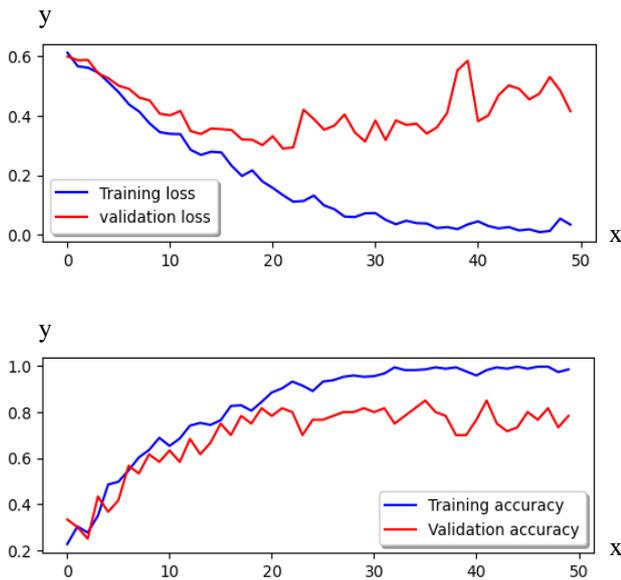
C. Training Model

Setelah model selesai dibuat, selanjutnya melakukan *training* model pada *dataset* penyakit kulit pada kucing. Proses *training* model menggunakan *fit* model dengan percobaan jumlah *epoch* sebanyak 30, 40, dan 50. *Epoch* merupakan jumlah berapa kali jaringan bekerja melewati seluruh *dataset* yang ada.

Tabel 5. Percobaan Jumlah *Epoch* Pada *Training* Model

Jumlah <i>Epoch</i>	Hasil Akurasi	Loss
30	0,55	0,65
40	0,80	0,66
50	0,78	0,41

Dari Tabel 5 di atas didapatkan hasil akurasi tertinggi berada pada *epoch* 50 dengan mendapatkan akurasi 0,78 atau 78%. Dari hasil tersebut, maka pada model yang dibuat menggunakan *epoch* 50. Setelah proses *trining* model selesai, maka akan didapatkan hasil berupa akurasi dan *loss* dari *training* dan validasi. Berikut merupakan hasil dari pelatihan model yang diperoleh dalam bentuk grafik yang dapat dilihat pada Gambar 8 di bawah ini.



Gambar 8. Grafik Hasil Training Model

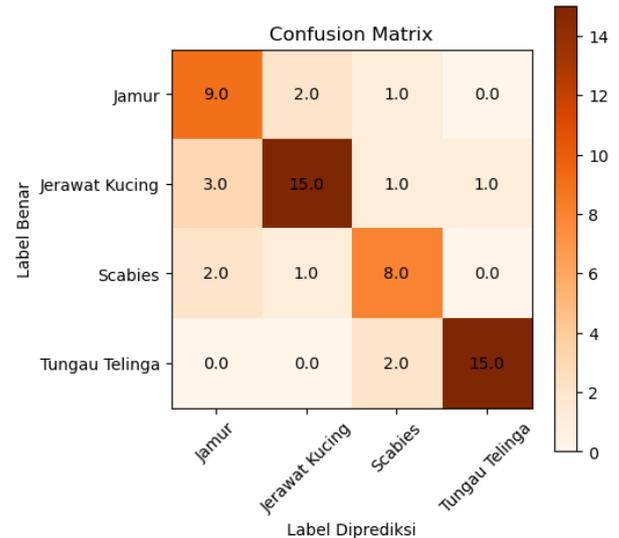
Pada grafik Gambar 8 di atas dapat dilihat jumlah *epoch* yang digunakan adalah 50 dengan akurasi rata-rata berada di nilai 0,7-0,8. Kemudian untuk *loss* masih sering terjadi peningkatan didalam proses *training* model dan berakhir pada nilai 0,41. Hasil *loss* yang dihasilkan sudah tergolong rendah karena berada dibawah nilai 0,5. *Loss* merupakan nilai yang menggambarkan kerugian dari semua kemungkinan yang dihasilkan oleh model yang sudah dibuat. *Loss* yang baik adalah menghasilkan nilai yang rendah. sedangkan akurasi merupakan representasi dari data uji yang diklasifikasikan ke kelas yang benar. Akurasi akhir yang didapatkan pada grafik diatas berada pada nilai 0,78 atau 78%. Artinya model yang dibangun sudah dikatakan baik, walaupun belum berada diatas 0,8. Apabila dilihat dari jumlah data yang digunakan yaitu 400 data gambar, model ini masih dapat melakukan deteksi gambar penyakit kulit pada kucing.

D. Pengujian dan Analisis

Pada pembuatan proyek akhir ini dilakukan beberapa pengujian terhadap hasil proyek akhir yang telah dikerjakan. Adapun pengujian yang digunakan pada proyek akhir ini adalah *black box testing*, uji analisis, dan *confusion matrix*.

a) *Confusion matrix*

Pada pengujian *confusion matrix*, jumlah *dataset* yang digunakan adalah 400 data gambar dengan pembagian menjadi data *training* dan *testing*. Pada data *training* ada sebanyak 340 data dimana masing-masing penyakit memiliki 85 data dan 60 data *testing* sebagai validasi memiliki masing-masing penyakit 15 data. Berikut merupakan hasil pengujian dari *confusion matrix* yang dapat dilihat pada Gambar 9 di bawah ini.



Gambar 9. Confusion Matrix

Dari Gambar 9 plot di atas, hasil *confusion matrix* untuk masing-masing kelas dapat dilihat pada Tabel 6 dibawah ini:

Tabel 6. Struktur Confusion Matrix

	Jamur	Jerawat Kucing	Scabies	Tungau Telinga
TP	9	15	8	15
FP	3+2+0 = 5	2+1+0 = 3	1+1+2 = 4	0+1+0 = 1
FN	2+1+0 = 3	3+1+1 = 5	2+1+0 = 3	0+0+2 = 2

Perhitungan nilai akurasi, presisi, dan *recall* yang sesuai pada Tabel 6 di atas:

Tabel 7. Hasil Perhitungan Confusion Matrix

Nilai	Nama Penyakit	Presentase
Akurasi	Jamur	64,2%
	Jerawat Kucing	83,33%
	Scabies	66,6%
	Tungau Telinga	93,75%
Presisi	Jamur	75%
	Jerawat Kucing	75%
	Scabies	61,53%
	Tungau Telinga	88,23%

Tabel 8. Hasil Rata-Rata Confusion Matrix

Parameter	Rata-rata
Akurasi	78,33%
Presisi	76,96%
Recall	74,94%

Dari hasil pengujian *confusion matrix* yang sudah dilakukan pada Tabel 7, kelas jamur, jerawat kucing, *scabies*, dan tungau telinga memiliki nilai TP (*True Positive*) lebih banyak, yang menandakan akurasi dari model sudah tinggi. Jumlah akurasi dari seluruh kelas adalah 78,33%. Untuk nilai *presisi* terendah berada pada kelas jamur dengan rata-rata nilai

64,2% dan untuk nilai *recall* terendah berada pada kelas *scabies* dengan rata-rata nilai 61,53%. Hal ini diakibatkan karena *dataset* yang digunakan saat proses *training* memiliki ciri-ciri penyakit yang kurang jelas. Untuk jamur hanya memperlihatkan area bulu rontok dan ada yang berbentuk lingkaran. Sedangkan untuk *scabies* memperlihatkan ciri-ciri berkerak yang secara sekilas mirip jamur karena gambar yang tidak begitu jelas.

Untuk kelas jerawat kucing dan tungau telinga menjadi kelas yang memiliki nilai presentasi paling tinggi dengan nilai rata-rata diatas 80%, dikarenakan *Dataset* yang digunakan lebih variasi dan bentuk ciri-ciri penyakit lebih kelihatan jelas. Hal ini membuat model tidak rancu dalam mendeteksi kedua kelas ini.

b) *Black Box Testing*

Pada pengujian ini dilakukan pada tiga *smartphone* dengan menjalankan aplikasi berdasarkan skenario yang sudah ditentukan. Berikut Tabel 9 yang merupakan hasil dari pengujian *black box testing* pada aplikasi.

Tabel 9. *Black Box Testing*

Skenario Uji	Hasil	Keterangan		
		I	II	III
Membuka aplikasi	Menampilkan <i>splash screen</i> berupa logo aplikasi	[√]	[√]	[√]
Mengklik menu tentang	Menampilkan halaman tentang aplikasi	[√]	[√]	[√]
Mengklik menu deteksi	Menampilkan halaman deteksi penyakit	[√]	[√]	[√]
Mengklik <i>button</i> kamera	Menampilkan halaman kamera dan menangkap gambar	[√]	[√]	[√]
Mengklik <i>button</i> galeri	Menampilkan halaman galeri dan <i>upload</i> gambar dari galeri	[√]	[√]	[√]
Mengklik <i>button</i> deteksi penyakit	Menampilkan halaman hasil dan hasil deteksi	[√]	[√]	[√]
Mengklik <i>button</i> kembali	Menampilkan halaman deteksi (halaman sebelumnya)	[√]	[√]	[√]
Mengklik <i>button</i> <i>back</i> <i>action bar</i>	Kembali ke halaman sebelumnya	[√]	[√]	[√]

c) Uji Analisis

Uji analisis dilakukan untuk menentukan pengaruh dari beberapa kondisi terhadap hasil akurasi dan deteksi pada

aplikasi. Perbandingan ini dilakukan dengan tujuan untuk mendapatkan kesimpulan dari pengujian.

1) Jarak dan Pencahayaan

Parameter uji untuk pengujian adalah jarak penyakit kulit pada kucing dengan kamera belakang *smartphone* dan dua kondisi pencahayaan, yaitu terang dan kurang cahaya. *Smartphone* yang digunakan memiliki kualitas kamera belakang dan spesifikasi yang berbeda-beda. Pengujian dilakukan kepada empat jenis penyakit kulit pada kucing dengan pengambilan lima gambar pada setiap jenis penyakit dan tiga jenis jarak, yaitu 5 cm, 10 cm, dan 20 cm. Selain jarak, pencahayaan juga dilakukan pengujian mulai dari pencahayaan terang serta kurang pencahayaan. Berikut spesifikasi beberapa *smartphone* yang digunakan dalam pengujian analisis yang dapat dilihat pada Tabel 10 di bawah.

Tabel 10. Resolusi Kamera *Smartphone*

No	<i>Smartphone</i>	Resolusi Kamera (Mega Pixel)
1.	<i>Smartphone</i> A (Xiaomi Redmi 9T)	42 MP
2.	<i>Smartphone</i> B (Xiaomi Redmi 10C)	50 MP
3.	<i>Smartphone</i> C (OPPO A52)	12 MP

Tabel 11. Rata-Rata Hasil Pengujian Pencahayaan Terang

<i>Smartphone</i>	Resolusi Kamera	Jarak			Rata-Rata akurasi
		5 cm	10 cm	20 cm	
<i>Smartphone</i> A	42 MP	70%	85%	40%	61,6%
<i>Smartphone</i> B	50 MP	80%	90%	45%	71,7%
<i>Smartphone</i> C	12 MP	80%	85%	50%	69,95%
Rata-Rata		73,3%	86,6%	45%	67,75%

Tabel 12. Rata-Rata Hasil Pengujian Pencahayaan Kurang

<i>Smartphone</i>	Resolusi Kamera	Jarak			Rata-Rata akurasi
		5 cm	10 cm	20 cm	
<i>Smartphone</i> A	42 MP	70%	80%	60%	69,95%
<i>Smartphone</i> B	50 MP	55%	55%	50%	53,3%
<i>Smartphone</i> C	12 MP	40%	35%	25%	33,33%
Rata-Rata		55%	56,6%	45%	52,193%

Dari hasil pengujian pada Tabel 11 dan 12 di atas, dapat diperoleh beberapa fakta-fakta sebagai berikut:

- Pada pencahayaan terang jarak 10 cm memiliki rata-rata akurasi tertinggi sebesar 86,6%
- Pada pencahayaan terang jarak 20 cm memiliki rata-rata akurasi terendah sebesar 45%
- Pada pencahayaan kurang terang jarak 10 cm memiliki rata-rata akurasi tertinggi sebesar 56,6%
- Pada pencahayaan kurang terang jarak 20 cm memiliki rata-rata akurasi terendah sebesar 45%
- *Smartphone* B memiliki rata-rata akurasi tertinggi sebesar 71,7% pada pencahayaan terang
- *Smartphone* A memiliki rata-rata akurasi terendah sebesar 65% pada pencahayaan terang
- *Smartphone* A memiliki rata-rata akurasi tertinggi sebesar 69,95% pada pencahayaan kurang terang

- *Smartphone C* memiliki rata-rata akurasi terendah sebesar 33,33% pada pencahayaan kurang terang

Dari fakta-fakta diatas, dapat disimpulkan bahwa semakin dekat dan semakin jauh jarak antara objek penyakit kulit pada kucing dengan kamera maka semakin rendah akurasi atau salah dalam mendeteksi penyakit kulit pada kucing Jarak terbaik untuk mendeteksi penyakit kulit pada kucing adalah 10 cm. Resolusi kamera belakang pada *smartphone* juga dapat mempengaruhi proses deteksi dan akurasi yang dihasilkan. Semakin dekat jarak maka gambar yang dihasilkan akan menjadi blur dan semakin jauh jarak maka ketajaman gambar akan berkurang sehingga dapat mempengaruhi akurasi dan hasil deteksi penyakit kulit pada kucing.

2) Kamera dan Upload

Setelah pengujian berdasarkan pengambilan jarak tangkap gambar dan pencahayaan, selanjutnya dilakukan pengujian untuk melihat konsistensi deteksi dan kualitas gambar dari aplikasi ketika mendeteksi penyakit kulit secara langsung pada kucing dengan kamera dan gambar yang di *upload*. Pengujian ini dilakukan sebanyak 5 kali untuk setiap *smartphone*. Berikut adalah hasil yang didapatkan dari pengujian yang sudah dilakukan.

Tabel 13. Hasil Deteksi Langsung dan Upload Smartphone

A		
No	Langsung	Upload
1	Benar	Benar
2	Benar	Benar
3	Salah	Benar
4	Salah	Salah
5	Benar	Benar

Tabel 14. Hasil Deteksi Langsung dan Upload Smartphone B

No	Langsung	Upload
1	Benar	Benar
2	Salah	Salah
3	Benar	Benar
4	Benar	Benar
5	Salah	Benar

Tabel 15. Hasil Deteksi Langsung dan Upload Smartphone C

No	Langsung	Upload
1	Salah	Salah
2	Salah	Benar
3	Benar	Benar
4	Benar	Benar
5	Salah	Salah

Dari seluruh percobaan yang sudah dilakukan pada tiga *smartphone* didapatkan hasil bahwa hasil deteksi ketika melakukan tangkap gambar dengan kamera langsung dengan gambar yang di *upload* dari galeri dominan mendapatkan hasil deteksi yang sama. Walaupun ada beberapa kali terjadi perbedaan hasil deteksi. Hal ini dapat disimpulkan bahwa citra

yang di-*upload* lebih banyak benar dibandingkan dengan tangkap kamera langsung, hal ini dikarenakan citra yang ditangkap langsung dengan kamera akan menyesuaikan dengan ukuran pada aplikasi dan berbentuk persegi. Ini akan membuat citra akan menjadi agak gepeng. Sedangkan ketika *upload* gambar dari galeri, kita bisa melakukan tangkap gambar dengan kamera dengan pengaturan hasil gambar 1:1 atau berbentuk persegi. Jadi ketika di-*upload* kedalam aplikasi citra tidak akan gepeng. Dengan demikian hasil deteksi pada aplikasi sudah cukup konsisten pada gambar yang sama walaupun dengan cara deteksi yang berbeda (kamera dan *upload*).

V. KESIMPULAN DAN SARAN

Dari hasil pengembangan dan penelitian yang sudah dilakukan pada aplikasi pendeteksi penyakit kulit pada kucing berbasis android didapatkan kesimpulan bahwa model yang dibuat sudah dapat mendeteksi penyakit kulit pada kucing. Akurasi deteksi penyakit dipengaruhi oleh jarak pengambilan objek penyakit dan resolusi kamera *smartphone*. Selain itu pencahayaan saat pengambilan gambar juga dapat mempengaruhi akurasi deteksi penyakit. Ketika objek lebih gelap atau kurang pencahayaan, akurasi menjadi menurun. Dari hasil pengujian pada setiap *smartphone* didapatkan rata-rata akurasi diantara 65% - 71,7% untuk pencahayaan terang. Dan untuk pencahayaan minim atau kurang terang didapatkan rata-rata akurasi model antara 33,33% - 69,95%. Hal ini menunjukkan bahwa pencahayaan dapat mempengaruhi akurasi deteksi penyakit kulit pada kucing. Untuk pengujian *confusion matrix* didapatkan akurasi klasifikasi benar saat pengujian sebesar 78,33%, presisi 76,96%, dan *recall* 74,94%. Dari hasil *confusion matrix* tersebut menunjukkan bahwa model sudah memiliki akurasi deteksi hampir mendekati 80% dan sudah mampu mendeteksi penyakit kulit pada kucing.

Saran untuk penilitan selanjutnya yang ingin mengembangkan sistem ini adalah ketika dalam mengumpulkan *dataset* dapat melakukan penambahan jumlah dan kualitas citra untuk meningkatkan akurasi pada model. *Dataset* yang dikumpulkan harus memiliki lebih banyak variasi seperti posisi citra, intensitas cahaya yang terang maupun kurang terang agar model lebih banyak belajar variasi citra dan dapat mendeteksi penyakit dari segala arah serta kondisi cahaya. Dan untuk pengembangan model CNN yang sudah dibuat pada penelitian ini dapat lebih dikembangkan dengan menambahkan arsitektur seperti *VGGNet*, *ResNet*, *Inception*, dan *DenseNet*. Hal ini bertujuan untuk meningkatkan kemampuan model dalam mempelajari fitur yang lebih kompleks dan mencegah *overfitting*.

REFERENSI

[1] I. Sukma and M. Petrus, "Sistem pakar penyakit kucing Menggunakan metode forward chaining Berbasis web," *Simtek: jurnal sistem informasi dan teknik komputer*, vol. 5, no. 1, pp. 52-58, 2020. doi:10.51876/simtek.v5i1.73.

-
- [2] M. Bela Pramesthi Putri and E. Santoso, "Diagnosis Penyakit Kulit Pada Kucing Menggunakan Metode Modified K-Nearest Neighbor," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 1, no. 12, pp. 1797–1803, 2017, [Online]. Available: <http://j-ptiik.ub.ac.id>
- [3] I. Gunaawan and Y. Fernando, "Sistem Pakar Diagnosa Penyakit Kulit Kucing Menggunakan Metode Naive Bayes Berbasis Web," *J. JTik (Jurnal Teknol. Inf. dan Komunikasi)*, vol. 2, no. 2, pp. 239–247, 2021, doi: 10.35870/jtik.v6i3.466.
- [4] I. W. Prastika, E. Zuliarso, J. T. Lomba, J. No, and S. 50241, "Deteksi Penyakit Kulit Wajah Menggunakan Tensorflow Dengan Metode Convolutional Neural Network," *J. Manaj. Inform. Sist. Informasi*, vol. 4, no. 2, pp. 84–91, 2021, [Online]. Available: <http://e-journal.stmiklombok.ac.id/index.php/misi>
- [5] M. A. Hanin, R. Patmasari, and R. Y. Nur, "Sistem Klasifikasi Penyakit Kulit Menggunakan Convolutional Neural Network (Cnn) Skin Disease Classification System Using Convolutional Neural Network (Cnn)," *e-Proceeding Eng.*, vol. 8, no. 1, pp. 273–281, 2021.
- [6] Y. Muflihah and G. Pramana, "Diagnosa penyakit kulit pada hewan kucing menggunakan metode forward chaining," *Jurnal Informasi dan Teknologi*, vol. 5, no. 3, pp. 51–57, 2023. doi:10.60083/jidt.v5i3.388.
- [7] A. Peryanto, A. Yudhana, and R. Umar, "Rancang Bangun Klasifikasi Citra Dengan Teknologi Deep Learning Berbasis Metode Convolutional Neural Network," *Format J. Ilm. Tek. Inform.*, vol. 8, no. 2, p. 138, 2020, doi: 10.22441/format.2019.v8.i2.007.
- [8] P. A. Nugroho, I. Fenriana, and R. Arijanto, "Implementasi Deep Learning Menggunakan Convolutional Neural Network (Cnn) Pada Ekspresi Manusia," *Algor*, vol. 2, no. 1, pp. 12–21, 2020.
- [9] A. Santoso and G. Ariyanto, "Implementasi Deep Learning berbasis Keras untuk Pengenalan Wajah," *Emit. J. Tek. Elektro*, vol. 18, no. 1, pp. 15–21, 2018, doi: 10.23917/emitor.v18i01.6235.
- [10] L. Agustien, T. Rohman, and A. W. Hujairi, "Real-time Deteksi Masker Berbasis Deep Learning menggunakan Algoritma CNN YOLOv3," *J. Teknol. Inf. dan Terap.*, vol. 8, no. 2, pp. 129–137, 2021, doi: 10.25047/jtit.v8i2.246.
- [11] S. Ilahiyah and A. Nilogiri, "Implementasi Deep Learning Pada Identifikasi Jenis Tumbuhan Berdasarkan Citra Daun Menggunakan Convolutional Neural Network," *JUSTINDO (Jurnal Sist. dan Teknol. Inf. Indones.)*, vol. 3, no. 2, pp. 49–56, 2018.
- [12] F. N. Cahya, N. Hardi, D. Riana, and S. Hadiyanti, "Klasifikasi Penyakit Mata Menggunakan Convolutional Neural Network (CNN)," *Sistemasi*, vol. 10, no. 3, p. 618, 2021, doi: 10.32520/stmsi.v10i3.1248.
- [13] H. A and M. I, "Penerapan Metode Convolutional Neural Network Pada Pengenalan Pola Citra Sandi Rumput," thesis, Universitas Komputer Indonesia, Bandung, 2019.
- [14] Y. D. Wijaya and M. W. Astuti, "Pengujian Blackbox Sistem Informasi Penilaian Kinerja Karyawan Pt Inka (Persero) Berbasis Equivalence Partitions," *J. Digit. Teknol. Inf.*, vol. 4, no. 1, p. 22, 2021, doi: 10.32502/digital.v4i1.3163.
-