

**TESIS**

**KLASIFIKASI CITRA KUCING DENGAN MENGGUNAKAN  
DEEP LEARNING: CONVOLUTION NEURAL NETWORK (CNN)**



Disusun oleh:

**Nama : Kumara Dewi Linda S.R**  
**NIM : 22.55.1214**  
**Konsentrasi : Digital Transformation Intelligence**

**PROGRAM STUDI S2 INFORMATIKA  
PROGRAM PASCASARJANA UNIVERSITAS AMIKOM YOGYAKARTA  
YOGYAKARTA**

**2024**

**TESIS**

**KLASIFIKASI CITRA KUCING DENGAN MENGGUNAKAN  
DEEP LEARNING: CONVOLUTION NEURAL NETWORK (CNN)**

**IMAGE CLASSIFICATION OF CAT USING DEEP LEARNING:  
CONVOLUTION NEURAL NETWORK (CNN)**

Diajukan untuk memenuhi salah satu syarat memperoleh derajat Magister



Disusun oleh:

**Nama : Kumara Dewi Linda S.R**  
**NIM : 22.55.1214**  
**Konsentrasi : Digital Transformation Intelligence**

**PROGRAM STUDI S2 INFORMATIKA  
PROGRAM PASCASARJANA UNIVERSITAS AMIKOM YOGYAKARTA  
YOGYAKARTA**

**2024**

**HALAMAN PENGESAHAN**

**KLASIFIKASI CITRA KUCING DENGAN MENGGUNAKAN  
DEEP LEARNING: CONVOLUTION NEURAL NETWORK (CNN)**

**IMAGE CLASSIFICATION OF CAT USING DEEP LEARNING: CONVOLUTION  
NEURAL NETWORK (CNN)**

Dipersiapkan dan Disusun oleh

**Kumara Dewi Linda S.R**

**22.55.1214**

Telah Diujikan dan Dipertahankan dalam Sidang Ujian Tesis  
Program Studi S2 INFORMATIKA  
Program Pascasarjana Universitas AMIKOM Yogyakarta  
pada hari Senin, 3 Juni 2024

Tesis ini telah diterima sebagai salah satu persyaratan  
untuk memperoleh gelar Magister Komputer

Yogyakarta, 3 Juni 2024

**Rektor**

**Prof. Dr. M. Suvanto, M.M.**

**NIK. 190302001**

**HALAMAN PERSETUJUAN**

**KLASIFIKASI CITRA KUCING DENGAN MENGGUNAKAN  
DEEP LEARNING: CONVOLUTION NEURAL NETWORK (CNN)**

**IMAGE CLASSIFICATION OF CAT USING DEEP LEARNING: CONVOLUTION  
NEURAL NETWORK (CNN)**

Dipersiapkan dan Disusun oleh

**Kumara Dewi Linda S.R**

**22.55.1214**

Telah Ditujikan dan Dipertahankan dalam Sidang Ujian Tesis  
Program Studi S2 INFORMATIKA  
Program Pascasarjana Universitas AMIKOM Yogyakarta  
pada hari Senin, 3 Juni 2024

**Pembimbing Utama**

**Prof. Dr. Kusriani, M.Kom.**  
**NIK. 190302106**

**Pembimbing Pendamping**

**Anggit Dwi Hartanto, M.Kom.**  
**NIK. 190302163**

**Anggota Tim Penguji**

**Alva Hendi M, S.T., M.Eng., Ph.D.**  
**NIK. 190302493**

**Tonny Hidayat, M.Kom., Ph.D.**  
**NIK. 190302182**

**Prof. Dr. Kusriani, M.Kom.**  
**NIK. 190302106**

Tesis ini telah diterima sebagai salah satu persyaratan  
untuk memperoleh gelar Magister Komputer

Yogyakarta, 3 Juni 2024

**Direktur Program Pascasarjana**

**Prof. Dr. Kusriani, M.Kom.**  
**NIK. 190302106**

## HALAMAN PERNYATAAN KEASLIAN TESIS

Yang bertandatangan di bawah ini,

Nama mahasiswa : Kumara Dewi Linda S.R  
NIM : 22.55.1214  
Konsentrasi : Digital Transformation Intelligence

Menyatakan bahwa Tesis dengan judul berikut:  
**Klasifikasi Citra Kucing Dengan Menggunakan Deep Learning: Convolution Neural Network (CNN)**

Dosen Pembimbing Utama : Prof. Dr. Kusriani, M.Kom.  
Dosen Pembimbing Pendamping : Anggit Dwi Hartanto, M.Kom

1. Karya tulis ini adalah benar-benar ASLI dan BELUM PERNAH diajukan untuk mendapatkan gelar akademik, baik di Universitas AMIKOM Yogyakarta maupun di Perguruan Tinggi lainnya
2. Karya tulis ini merupakan gagasan, rumusan dan penelitian SAYA sendiri, tanpa bantuan pihak lain kecuali arahan dari Tim Dosen Pembimbing
3. Dalam karya tulis ini tidak terdapat karya atau pendapat orang lain, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan disebutkan dalam Daftar Pustaka pada karya tulis ini
4. Perangkat lunak yang digunakan dalam penelitian ini sepenuhnya menjadi tanggung jawab SAYA, bukan tanggung jawab Universitas AMIKOM Yogyakarta
5. Pernyataan ini SAYA buat dengan sesungguhnya, apabila di kemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka SAYA bersedia menerima SANKSI AKADEMIK dengan pencabutan gelar yang sudah diperoleh, serta sanksi lainnya sesuai dengan norma yang berlaku di Perguruan Tinggi

Yogyakarta, 3 Juni 2024  
Yang Menyatakan,



Kumara Dewi Linda S.R

## HALAMAN PERSEMBAHAN

Dengan segala puji dan syukur kepada Allah SWT, yang telah memberikan rahmat, karunia dan petunjuk sehingga penulis dapat menyelesaikan penelitian dengan judul “Klasifikasi Citra Kucing Dengan Menggunakan *Deep Learning: Convolution Neural Network (CNN)*” dengan baik. Oleh karena itu penulis ingin mengucapkan terimakasih kepada:

Ibu Prof. Dr. Kusriani, M.Kom dan Bapak Anggit Dwi Hartanto, M.Kom selaku dosen pembimbing atas bimbingan dan arahan selama proses penelitian ini.

Keluarga, suami dan anak yang *senantiasa* memberikan doa dan dukungan selama proses penelitian ini.

Rekan-rekan penelitian atas sharing dan diskusi dalam penelitian ini.

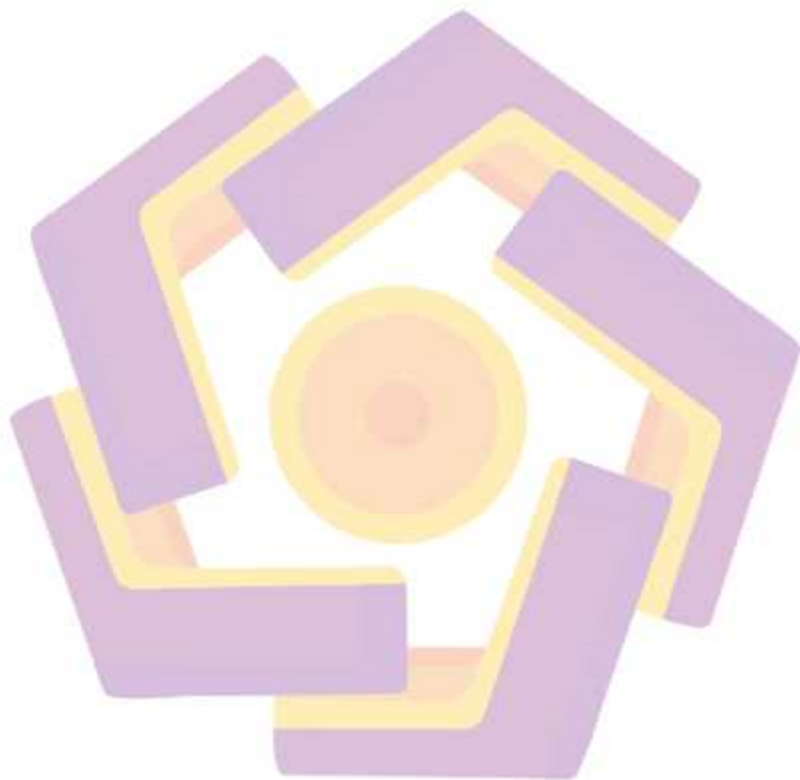
Terimakasih

Yogyakarta, 3 Juni 2024

Penulis

## HALAMAN MOTTO

Allah tidak membebani seseorang diluar kemampuannya (Al-Baqarah: 286)





## KATA PENGANTAR

Puji Syukur kami panjatkan kehadirat Allah SWT, yang telah memberikan Rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan penelitian yang berjudul “Klasifikasi Citra Kucing Dengan Menggunakan *Deep Learning: Convolution Neural Network (CNN)*”.

Penelitian ini merupakan upaya penulis untuk dapat mendalami klasifikasi citra, khususnya pada pengolahan citra kucing. Seiring perkembangan teknologi, pengolahan citra menjadi aspek sangat penting dalam klasifikasi.

Algoritma *Convolutional Neural Network (CNN)* adalah bagian dari *deep learning network* yaitu jenis jaringan saraf tiruan yang saat ini banyak digunakan untuk pengenalan suatu citra. Dalam penelitian ini, citra kucing diambil sebagai dataset untuk memahami implementasi penggunaan algoritma *Convolutional Neural Network (CNN)* dalam klasifikasi citra.

Penelitian ini diharapkan dapat memberikan kontribusi positif dan pemahaman yang mendalam mengenai implementasi algoritma *Convolutional Neural Network (CNN)* dalam konteks pengelolaan citra. Selain itu, penelitian ini diharapkan mampu memberikan rekomendasi yang bermanfaat untuk perbaikan dan pengembangan proses klasifikasi citra.

Penelitian ini tidak terlepas dari dukungan berbagai pihak yang telah memberikan bimbingan, arahan dan motivasi. Oleh karena itu, penulis menyampaikan penghargaan yang setinggi-tingginya kepada:



**1. Dosen Pembimbing**

Atas bimbingan dan arahan yang sangat berharga dalam proses penelitian ini.

**2. Suami, anak, dan keluarga**

Atas doa, dukungan dan motivasi yang selalu diberikan selama proses penelitian ini

**3. Rekan-rekan penelitian**

Atas diskusi dan kerjasama sehingga memperluas pengetahuan dan pemahaman hasil penelitian.

Akhir kata, penulis menyadari bahwa penelitian ini masih jauh dari kata sempurna. Oleh karena itu, masukan dan saran yang bersifat membangun selalu diharapkan untuk pengembangan penelitian dimasa mendatang.

Terimakasih

Yogyakarta, 3 Juni 2024

Penulis

## DAFTAR ISI

HALAMAN JUDUL.....	ii
HALAMAN PENGESAHAN.....	iii
HALAMAN PERSETUJUAN.....	iv
HALAMAN PERNYATAAN KEASLIAN TESIS.....	v
HALAMAN PERSEMBAHAN.....	vi
HALAMAN MOTTO.....	vii
KATA PENGANTAR.....	viii
DAFTAR ISI.....	x
DAFTAR TABEL.....	xii
DAFTAR GAMBAR.....	xiii
INTISARI.....	xv
<i>ABSTRACT</i> .....	xvi
<b>BAB I PENDAHULUAN</b> .....	1
1.1. Latar Belakang Masalah.....	1
1.2. Rumusan Masalah.....	4
1.3. Batasan Masalah.....	4
1.4. Tujuan Penelitian.....	4
1.5. Manfaat Penelitian.....	5
<b>BAB II TINJAUAN PUSTAKA</b> .....	6
2.1. Tinjauan Pustaka.....	6
2.2. Keaslian Penelitian.....	10

2.3. Landasan Teori.....	18
<b>BAB III METODE PENELITIAN.....</b>	<b>34</b>
3.1. Jenis, Sifat, dan Pendekatan Penelitian.....	34
3.2. Metode Pengumpulan Data.....	34
3.3. Metode Analisis Data.....	35
3.4. Alur Penelitian.....	35
<b>BAB IV HASIL PENELITIAN DAN PEMBAHASAN.....</b>	<b>39</b>
4.1. Pengumpulan Data.....	39
4.2. Pre Processing.....	44
4.3. Pembuatan Model Arsitektur CNN.....	46
4.4. Analisis Data.....	49
4.5. Analisis Hasil Penelitian.....	50
<b>BAB V PENUTUP.....</b>	<b>62</b>
5.1. Kesimpulan.....	62
5.2. Saran.....	62
<b>DAFTAR PUSTAKA.....</b>	<b>63</b>
<b>LAMPIRAN.....</b>	<b>66</b>

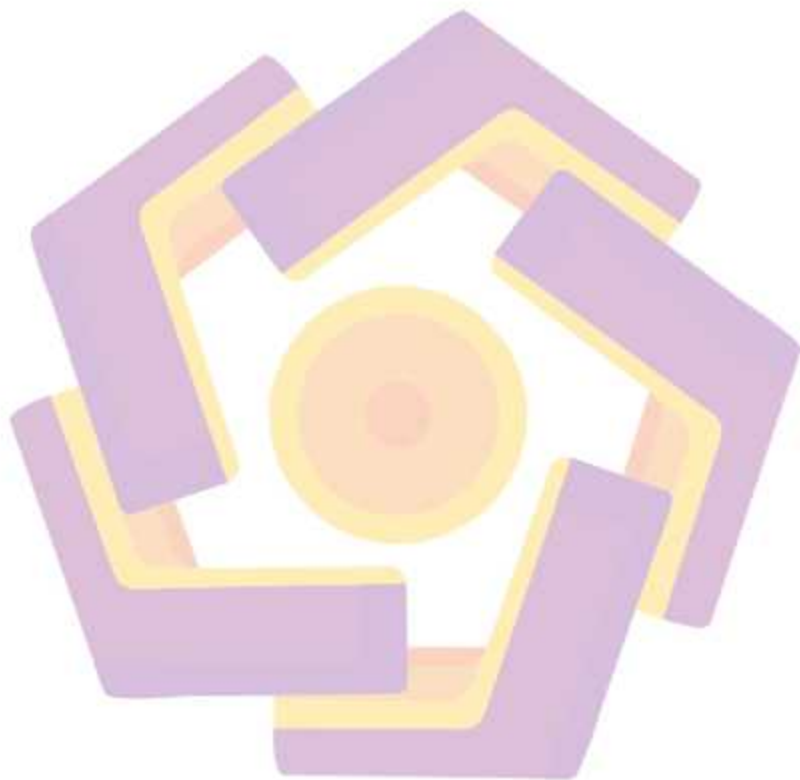
## DAFTAR TABEL

Tabel 2.1. Matriks literatur review.....	10
Tabel 2.2. Taksonomi Kucing.....	18
Table 4.1. Tabel contoh citra kucing.....	39
Tabel 4.3. Tabel hasil Model Dataset Satu.....	46
Tabel 4.4. Tabel Hasil Model Dataset Dua.....	47
Tabel 4.5 Hasil Percobaan Skenario Pertama Dataset Satu.....	50
Tabel 4.6. Hasil Percobaan Skenario Pertama Dataset Dua.....	54
Tabel 4.7. Perbandingan Akurasi Training dan Testing.....	57
Tabel 4.8. Precision, Recall dan F1 Score.....	58

## DAFTAR GAMBAR

Gambar 2.1. Machine Learning .....	21
Gambar 2.2. Arsitektur Sistem Data mining (Han dan Kamber, 2006).....	23
Gambar 2.3. Convolutional Neural Neteork (CNN) .....	26
Gambar 2.4. Lapisan Pada Algoritma CNN.....	27
Gambar 2.5. Ilustrasi filter pada algoritma CNN.....	28
Gambar 2.6. Ilustrasi 3D Convolution layer CNN.....	28
Gambar 2.7. Proses Klasifikasi.....	29
Gambar 2.8. Visualisasi Cara Kerja CNN.....	30
Gambar 2.9. Confusion Matrix.....	31
Gambar 3.1. Pengumpulan data.....	34
Gambar 3.2. Alur Penelitian.....	35
Gambar 4.1 Pelabelan dataset.....	38
Gambar 4.2 Workbook Phyton.....	41
Gambar 4.3. Rumus <i>RMSProp Optimizier</i> .....	42
Gambar 4.4 Jumlah data Latihan dan Validasi Dataset Satu.....	43
Gambar 4.5 Jumlah data Latihan dan Validasi Dataset Dua .....	44
Gambar 4.6 Menampilkan citra .....	44
Gambar 4.7 Tampilan Citra Kucing.....	45
Gambar 4.8. <i>Code</i> Augmentasi .....	48
Gambar 4.9 Grafik Akurasi Training dan Validasi Dataset Satu .....	53
Gambar 4.10. Grafik Akurasi Training dan Validasi Dataset Dua .....	57

Gambar 4.11. Confusion Matriks Dataset Satu .....	59
Gambar 4.12. Confusion Matriks Dataset Dua .....	60



## INTISARI

*Deep learning* merupakan bagian dari *machine learning* yang memiliki kemampuan untuk mengenali pola gambar, suara, teks dan data lainnya yang kompleks sehingga dapat menghasilkan prediksi yang akurat. Salah satu kemampuan *deep learning* adalah klasifikasi citra pada objek. *CNN* adalah salah satu metode dalam *machine learning* yang digunakan untuk mengklasifikasikan citra objek. Algoritma *Convolutional Neural Network (CNN)* adalah bagian dari *deep learning network* yaitu jenis jaringan saraf tiruan yang saat ini banyak digunakan untuk pengenalan suatu citra. Dalam penelitian ini, algoritma yang digunakan adalah *CNN* karena akurasinya yang cukup baik. *Deep learning* dengan *convolutional neural network (CNN)* yang banyak digunakan untuk melakukan deteksi, klasifikasi, dan prediksi pada gambar. Citra objek dalam penelitian ini adalah kucing yang terdiri dari berbagai macam jenis. Tujuan dari penelitian ini adalah untuk mengklasifikasikan citra kucing sesuai dengan jenisnya. Penelitian ini merupakan tinjauan literatur untuk menambah pengetahuan berharga mengenai penelitian terbaru tentang klasifikasi citra kucing menggunakan *CNN*. Penelitian ini membahas studi literatur tentang variabel input, metode yang digunakan dan hasil literatur dari penelitian sebelumnya. Metode yang paling banyak digunakan pada penelitian sebelumnya adalah *CNN*. Berdasarkan studi literatur yang telah dilakukan, maka metode *convolutional neural network* yang menghasilkan akurasi diatas 80 %.

Kata kunci: *Deep Learning; Klasifikasi citra kucing; Image Processing; CNN*



## **ABSTRACT**

*Deep learning is a part of machine learning that has the ability to recognize complex patterns of images, sounds, text and other data so that it can produce accurate predictions. One of the capabilities of deep learning is image classification of objects. CNN is a method in machine learning that is used to classify object images. The Convolutional Neural Network (CNN) algorithm is part of a deep learning network, namely a type of artificial neural network that is currently widely used for image recognition. In this research, the algorithm used is CNN because its accuracy is quite good. Deep learning with convolutional neural networks (CNN) is widely used to detect, classify and predict images. The object image in this research is a cat which consists of various types. The aim of this research is to classify cat images according to their type. This research is a literature review to add valuable knowledge regarding the latest research on cat image classification using CNN. This research discusses literature studies regarding input variables, the methods used and literature results from previous research. The method most widely used in previous research is CNN. Based on the literature study that has been carried out, the convolutional neural network method produces accuracy above 80%.*

*Keywords: Deep Learning, Cat Image Classification, Image Processing, CNN;*

# BAB I

## PENDAHULUAN

### 1.1. Latar Belakang Masalah

Kucing adalah salah satu binatang yang disukai oleh banyak orang. Bahkan banyak pecinta binatang yang menjadikan kucing sebagai binatang peliharaan favorit. Hal itu dikarenakan binatang tersebut lucu dan menggemaskan. Kucing memiliki daya tarik tersendiri karena warna bulu, bentuk tubuh, warna mata dan ras yang beraneka ragam. Tetapi seringkali pemilik kucing tidak mengetahui jenis kucing yang dimiliki. Dalam bahasa latin kucing disebut *Fell silveltris catus* yaitu sejenis karnivora. Kucing membutuhkan perawatan khusus yang sesuai dengan rasnya agar kesehatannya terjamin. Kucing ras adalah binatang yang tergolong pintar dan dapat menghibur karena tingkahlakunya.

Banyaknya jumlah jenis kucing yang diakui secara internasional yaitu 142 kucing seperti Persia, Birman, Sphynk. Kucing persia memiliki ciri hidung pesek, bertubuh agak membulat dan berbulu panjang. Karena banyaknya jumlah jenis maka dibutuhkan sebuah algoritma yang dapat mengklasifikasikan citra kucing agar pemilik kucing mengetahui jenis kucing apa yang dimilikinya. Dalam penelitian ini penulis menggunakan 5 ras kucing yaitu: *Abyssinian, Bengal, Birman, Persia, dan Sphynx*. Pemilihan ras kucing berdasarkan banyaknya pecinta kucing dengan ras tersebut, karena memiliki ciri-ciri yang khas seperti pada ras persia yang memiliki ciri tubuh gemuk dan hidung pesek, mata biru dan wajah bulat pada ras birman, sphynk yang memiliki ciri sangat menonjol yaitu tidak memiliki bulu, kucing

Bengal yang memiliki warna bulu seperti macan tutul dan ras Abyssinian yang mempunyai bulu yang lebat dan halus.

Algoritma *Convolutional Neural Network* (CNN) adalah bagian dari *deep learning network* yaitu jenis jaringan saraf tiruan yang saat ini banyak digunakan untuk pengenalan suatu citra. Dalam penelitian ini, algoritma yang digunakan adalah CNN karena akurasi yang cukup baik. *Deep learning* dengan *convolutional neural network* (CNN) yang banyak digunakan untuk melakukan deteksi, klasifikasi, dan prediksi pada gambar. Algoritma CNN memiliki *neuron* yang didesain untuk bekerja seperti *lobus frontal*, terutama pada bagian *visual cortex* pada manusia dan hewan. *Convolutional neural network* (CNN) adalah jenis jaringan syaraf tiruan khusus untuk memproses data yang dikenal memiliki *grid-like topologi*.

Berdasarkan acuan dari penelitian sebelumnya, penerapan CNN banyak digunakan pada kasifikasi citra. Beberapa penelitian sebelumnya antara lain, Alamsyah (2020) melakukan penelitian dalam mengklasifikasikan ekspresi citra wajah dengan algoritma CNN, (Fawwas dkk, 2021) tentang klasifikasi ras pada kucing menggunakan *Convolutional Neural Network* (CNN). Selain itu pada penelitian (Maylani dkk, 2021) mengenai implementasi metode *Data Mining* untuk Memprediksi Warna Anak Kucing Pada Proses Pengembangbiakan Kucing Ras Menggunakan Algoritma *Support Vector Machine* (SVM). Penerapan CNN di bidang lain diantaranya, dibidang kebudayaan yaitu untuk mendeteksi objek-objek pada bidang kebudayaan. Putra (2020) menerapkan algoritma CNN dalam mengklasifikasikan penyakit retinopati yang dapat menyebabkan kebutaan

terhadap mata. Arsitektur yang digunakan adalah *GoogLeNet*, *ResNet18*, *ResNet50*, dan *ResNet101*. Tahap-tahap dalam penelitian ini adalah preprosesing data, peningkatan kualitas citra, ekstraksi fitur, pengurangan fitur dan klasifikasi. Dataset citra sebesar 224 x 224 piksel yang terbagi menjadi 75 % sebagai data latih data 25 % sebagai data uji. Untuk meningkatkan kualitas citra, digunakan 3 jenis metode yang masing-masing akan diujikan, yaitu metode *CLAHE (Contrast Limited Adaptive Histogram Equalization)*, *Homomorphic filter*, dan *Morphological Contrast Enhancement*. Dari hasil percobaan yang telah dilakukan didapatkan hasil akurasi tertinggi yaitu pada arsitektur *ResNet50* untuk klasifikasi dataset kelas dengan nilai 86,76%.

Berdasarkan permasalahan pada penelitian sebelumnya, maka penulis tertarik untuk melakukan penelitian menggunakan algoritma *Convolutipnal Neural Network (CNN)* pada citra kucing sebagai objeknya. Algoritma CNN dipilih karena mempunyai kinerja terbaik dibandingkan algoritma yang lain berdasarkan penelitian-penelitian sebelumnya. Agar memperoleh tingkat akurasi yang lebih baik maka penulis menggunakan beberapa skenario yang terdiri dari lima kelas dengan jumlah dataset yang berbeda. Untuk meningkatkan nilai gradient yang terlalu kecil pada CNN agar dapat dihitung dengan cepat oleh CNN maka penulis menggunakan *Relu*. Kontribusi dari penelitian ini adalah algoritma CNN diharapkan dapat menghasilkan waktu komputasi yang lebih singkat pada klasifikasi citra kucing.



## 1.2. Rumusan Masalah

Berdasarkan latar belakang di atas, maka rumusan masalah pada penelitian ini adalah bagaimana membangun model sistem kalsifikasi dengan arsitektur yang diusulkan untuk mengenali citra kucing dan dapat dibandingkan performanya dengan penelitian sebelumnya.

## 1.3. Batasan Masalah

Agar penelitian dapat terlaksana dengan terarah, maka batasan masalah dalam penelitian sangat penting. Adapun batasan masalah pada penelitian ini adalah sebagai berikut:

- a. Dalam penelitian ini algoritma yang digunakan adalah CNN
- b. Data dalam penelitian ini adalah citra kucing
- c. Data objek citra kucing untuk proses training dan testing berasal dari *Oxford-IIIT*
- d. Ekstraksi fitur citra berdasarkan ciri-ciri bentuk dari objek citra kucing dengan algoritma CNN
- e. Dataset yang digunakan adalah dataset 5 ras kucing yang terdiri atas ras *Abbyssian, Bengal, Birman, Persia, dan Sphynx*.

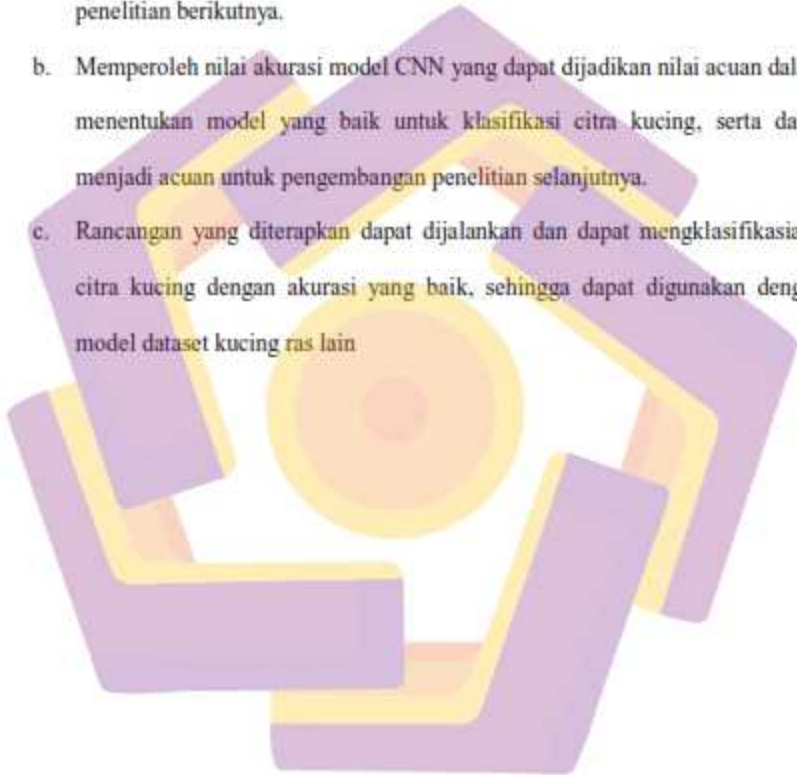
## 1.4. Tujuan Penelitian

Tujuan penelitian ini adalah untuk membangun sebuah model sistem dengan menggunakan algoritma CNN yang dapat mengenali citra kucing ras tertentu dan dapat mengukur performa model yang dihasilkan.

### 1.5. Manfaat Penelitian

Adapaun manfaat dari penelitian ini adalah :

- a. Memberikan model CNN yang dapat digunakan untuk mengklasifikasikan citra kucing yang terdapat pada dataset *Oxford-IIIT* dan untuk referensi pada penelitian berikutnya.
- b. Memperoleh nilai akurasi model CNN yang dapat dijadikan nilai acuan dalam menentukan model yang baik untuk klasifikasi citra kucing, serta dapat menjadi acuan untuk pengembangan penelitian selanjutnya.
- c. Rancangan yang diterapkan dapat dijalankan dan dapat mengklasifikasi citra kucing dengan akurasi yang baik, sehingga dapat digunakan dengan model dataset kucing ras lain



## BAB II

### TINJAUAN PUSTAKA

#### 2.1. Tinjauan Pustaka

Tinjauan pustaka dalam penelitian sangatlah penting, selain sebagai referensi tinjauan pustaka dilakukan untuk menghindari duplikasi atau *plagiarisme*. Berikut ini ulasan dari beberapa penelitian yang dijadikan referensi dalam penelitian ini.

Penelitian oleh Muhammad Afif Amanullah Fawwas, Kurniawan Nur Ramadhani, Febryanti Sthevanie (2020) mengenai klasifikasi ras kucing menggunakan algoritma *convolutional neural network* (CNN). Dalam penelitian ini bertujuan untuk mencari model yang terbaik antara *VGG16*, *InterceptionV3*, *ResNet50*, dan *Xception*. Evaluasi dilakukan dengan menggunakan metrik akurasi. Penelitian dilakukan dengan membandingkan ketiga parameter yang telah ditentukan yaitu tanpa menggunakan *transfer learning*, menggunakan *transfer learning* tanpa *fine tuning*, dan menggunakan *transfer learning* dengan *fine tuning*. Hasil testing didapatkan berupa akurasi untuk tiap modelnya yaitu 60,85%, 84,94%, 71,39%, dan 93,75%.

Penelitian lain yaitu Ferdy Maylani, Sriyanto (2021) tentang Implementasi Metode *Data Mining* Untuk Memprediksi Warna Anak Kucing Pada Proses Pengembangbiakan Kucing Ras Menggunakan Algoritma *Support Vector Machine* (SVM). Penelitian ini bertujuan untuk membuat prediksi warna anak kucing berdasarkan warna kedua induknya sehingga para peternak pemula dapat terbantu



untuk menentukan rencana perkawinan (*matting*) dengan lebih matang dan akurat serta para pengabdosi juga dapat lebih terbantu dalam memenuhi keinginannya memiliki ras kucing dengan warna tertentu. Dataset dibuat dengan menggunakan tiga atribut, yaitu warna ayah, warna induk, dan warna anak yang nantinya sebagai hasil prediksi. Tool yang digunakan dalam penelitian ini adalah *rapidminer* agar dapat melakukan perbandingan metode prediksi dengan tingkat akurasi tertinggi. Hasil penelitian ini menunjukkan bahwa tingkat akurasi yang tidak bisa tinggi sekali (diatas 90%) karena dipengaruhi oleh Gen Dominan atau Gen Resesif. Gen yang dominan dapat menghasilkan warna yang sama dengan mengubah hasil dari prediksi sebelumnya.

Dalam penelitian Muhammad Ridwan Effendi tentang sistem Deteksi Wajah Jenis Kucing dengan *image classification* menggunakan *OpenCV*. Penelitian ini menganalisa identifikasi dan mengenali jenis kucing Anggora dan Persia melalui bentuk hidung dan wajah dan kaki yang dapat diambil melalui *image* dengan menggunakan *OpenCV*. Metode yang digunakan dalam penelitian ini adalah berupa pengamatan langsung, pengumpulan data objek, dan studi pustaka (*Library Research*) dengan menggunakan perangkat lunak *Visual Studio* dan *library OpenCV* dan *Python*. Hasil dari analisis penelitian adalah metode *viola-jones* dan *Haarcascade* dapat diimplementasikan dalam pendeteksian mengenali bentuk wajah jenis kucing anggora dan persia.

Penelitian Naura Qutrunnada (2021) tentang klasifikasi ras kucing dengan *convolutional neural network*. Penelitian ini menggunakan 5 jenis ras kucing dengan setiap jenis memiliki 200-300 data penelitian. Hasil pengujian

menggunakan *confusion matrix* dan mendapatkan nilai *presisi*, *recall*, *f1 score* dan *akurasi* sebesar 100% pada citra multi objek dengan 2 objek dan 3 objek. Pada citra dengan 4 objek mendapatkan nilai *presisi*, *recall*, *f1 score* dan *akurasi* masing-masing sebesar 89%, 87%, 87% dan 95%. Sedangkan nilai *presisi*, *recall*, *f1 score* dan *akurasi* pada citra dengan 5 objek masing-masing mendapatkan 87%, 86%, 86% dan 94%.

Penelitian A. Slamet Riyadi (2021) tentang klasifikasi citra anjing dan kucing menggunakan metode Convolutional Neural Network (CNN) bertujuan untuk mendeteksi objek citra kucing dan anjing, serta membedakan model citra antara kedua objek citra. Hasil dari penelitian tersebut adalah aplikasi program klasifikasi berbasis CNN dengan hasil pengujian pada sistem dan aplikasi CNN digunakan untuk mengenali objek anjing dan kucing dan capaian akurasi dan presisi sistem klasifikasi terhadap pengenalan anjing dan kucing yang didapat yaitu sebesar 84,09%.

Penelitian Teguh Ardiyanto (2022) mengenai Classification of dog and cat images using the CNN method yang bertujuan untuk Mengklasifikasikan citra anjing dan kucing menggunakan algoritma CNN. Kesimpulan dari peneliti tersebut adalah CNN belum dapat bekerja secara maksimal. Hal ini terbukti setelah sistem dianalisis menggunakan Confusions Matric dengan presisi 45%, recall 45%, dan f1-score 45%. Tingkat akurasi, presisi, recall dan f1-score kecil, oleh karena itu perlu adanya perbaikan pada sistem.

Penelitian selanjutnya adalah *Image Classification – Cat and Dog Images* yang dilakukan oleh Tushar Jajodia. Masalah utamanya adalah klasifikasi gambar,

yang didefinisikan sebagai memprediksi kelas dari gambar tersebut dengan menghasilkan sebuah jaringan saraf konvolusi dalam mendalam untuk klasifikasi gambar (gambar kucing dan anjing). Meskipun hanya menggunakan subset dari gambar-gambar tersebut, berhasil mencapai akurasi sebesar 90,10%.

Nadia Azahro (2022) deteksi ras kucing menggunakan *compound model scaling convolutional neural network*. Penelitian ini bertujuan untuk mengklasifikasikan kucing berdasarkan ras menggunakan *CNN*. Hasil dari penelitian ini adalah arsitektur *EfficientNet-B0* sebagai base model mendapatkan akurasi yang paling tinggi yaitu sebesar 98% menggunakan *optimizer Adam* dan *RMSprop* dengan *learning rate* 0.001. Namun masih terjadi *overfitting* pada model ini. Model paling optimal mendapatkan akurasi sebesar 95% dan akurasi validasi sebesar 91%. Model ini berhasil mengklasifikasikan 179 dari 180 gambar kucing dengan kelas yang benar. Model ini yaitu menggunakan *optimizer RMSprop* dengan *learning rate* 0.0001 dan menggunakan data *pre-processing*. Semakin kecil *learning rate* semakin kecil kemungkinan terjadinya *overfitting*, namun akurasi yang didapatkan juga lebih rendah.

## 2.2. Keaslian Penelitian

Tabel 2.1. Matriks Literatur Review

Klasifikasi Citra Kucing Menggunakan Algoritma *Convolutional Neural Network (CNN)*

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
1.	Klasifikasi Ras pada Kucing menggunakan Algoritma <i>Convolutional Neural Network (CNN)</i>	Muhammad Afif Amanullah Fawwaz1, Kurniawan Nur Ramadhani, S.T., M.T.2, Febryanti Sthevanie, S.T., M.T.3, e- <i>Proceeding of Engineering, 2021</i>	Penelitian kali ini bertujuan untuk mencari model yang terbaik antara <i>VGG16, InceptionV3, ResNet50</i> , dan <i>Xception</i> . Evaluasi dilakukan dengan menggunakan metrik akurasi.	<i>Model Xception + Transfer Learning + Fine Tuning</i> adalah model yang terbaik. Model ini menghasilkan akurasi, precision, recall, dan <i>f1-score</i> dengan nilai 93.75%, 93.74%, 93.56%, dan 93.64%.	Pada saat proses training dilakukan sebaiknya menggunakan perangkat dengan spesifikasi yang mumpuni, karena model-model yang diuji memberikan beban komputasi yang besar.	Dalam penelitian ini algoritma yang akan digunakan sama akan tetapi jumlah data set dan kelas berbeda
2.	Implementasi Metode <i>Data Mining</i> Untuk Memprediksi Warna Anak Kucing Pada Proses pengembangbiakan Kucing Ras menggunakan	Ferdy Maylani1, Sriyanto, Nosiell, Seminar Nasional Hasil Penelitian dan Pengabdian Masyarakat, 2021	Untuk membantu para peternak kucing ras dalam memprediksi warna anak kucing sehingga dapat memenuhi permintaan para pengadopsi yang	Penggunaan <i>Data Mining</i> dalam memprediksi warna anak kucing berdasarkan warna kedua orang tuanya bisa menjadi solusi untuk paling tidak mempermudah para peternak kucing yang belum bahkan tidak	Karena keterbatasan waktu hanya berupa prediksi warna yang dapat penulis sampaikan. Kedepannya penulis	Penelitian ini menggunakan Algoritma SVM, dan penulis menggunakan CNN dengan sama-sama menggunakan citra kucing



Tabel 2.1. (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
	<i>Algoritma Support Vector Machine (SVM)</i>		menginginkan anak kucing ras dengan warna tertentu.	menguasi ilmu genetika. Tingkat akurasi yang tidak bisa tinggi sekali (diatas 90%) karena dipengaruhi oleh Gen Dominan atau Gen Resesif. Gen yang dominan mungkin dapat menghasilkan warna yang sama dengan mengubah hasil dari prediksi sebelumnya.	harapkan dapat mengumpulkan dataset yang lebih banyak dan juga dapat meningkatkan prediksi tidak hanya berupa warna tetapi juga dapat berupa pola warna beserta jenis kelamin anak kucingnya.	
3.	Sistem Deteksi Wajah Jenis Kucing Dengan <i>Image Classification</i> Menggunakan <i>OpenCV</i>	Muhammad Ridwan Effendi	Menganalisa identifikasi dan mengenali jenis kucing Anggora dan Persia	<ol style="list-style-type: none"> <li>1. Metode <i>Viola-Jones</i> dan <i>Haar Cascade</i> dapat diimplementasikan dalam pendeteksian mengenali bentuk wajah jenis kucing Anggora dan persia.</li> <li>2. <i>Image</i> sebagai objek penulis <i>capture</i> gambar yang ingin di olah di <i>OpenCV</i> lalu</li> </ol>	Menambah jenis ras kucing dalam penelitian agar perbandingan tingkat akurasi lebih baik	Menambah jenis ras kucing, penggunaan metode yang lain untuk mengetahui metode apa yang sesuai untuk penyelesaian masalah dalam penelitian ini

Tabel 2.1. (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
				3. gambar akan dibandingkan di <i>dataset</i> dengan LBPH jika ada yang mendekati maka akan menampilkan hasil gambar deteksi. Sistem dapat mendeteksi adanya beberapa bentuk wajah kucing dalam suatu citra.		
4.	Klasifikasi Ras Kucing Dengan <i>Convolutional Neural Network</i> Pada Citra Multi Objek	Naura Qatrunnada, M. Fachrurrozi, Alvi Syahrini Utami Sriwijaya Journal of Informatic and Applications Vol. 3, No. 2, August 2022, pp. 74-84 ISSN 2807-2391	1. Mengimplementasikan metode CNN untuk klasifikasi ras kucing pada citra multi objek 2. Mengetahui nilai <i>presisi, recall, f score</i> dan akurasi yang dihasilkan metode CNN dalam penerapannya untuk klasifikasi	1. Klasifikasi ras kucing pada beberapa objek gambar dapat dilakukan dengan menggunakan CNN dengan Arsitektur <i>Xception</i> . 2. Implementasi CNN pada citra multi objek menghasilkan nilai <i>presisi, recall, skor f1</i> dan akurasi 100% pada gambar dengan 2 objek dan 3 objek.	Menambah arsitektur CNN untuk mengetahui hasil akurasi yang lebih baik	Pada penelitian terdahulu menggunakan citra lebih dari satu, sedangkan pada penelitian ini menggunakan 1 objek tetapi dengan jumlah skenario 2.

Tabel 2.1. (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
			ras kucing pada citra multi objek.	<p>nilai, <i>recall</i>, <i>skor f1</i> dan akurasi masing-masing sebesar 89%, 87%, 87% dan 95%. Sedangkan presisinya nilai, <i>recall</i>, <i>skor f1</i> dan akurasi pada citra dengan 5 objek memperoleh nilai 87%, 86%, 86% dan 94%.</p> <p>3. Semakin banyak objek dalam suatu gambar, maka nilai akurasinya akan mulai berkurang. Dan sebaliknya, semakin sedikit objek maka nilai akurasinya akan semakin meningkat. Waktu komputasi juga dipengaruhi oleh angka objek yang terdapat dalam gambar. Semakin banyak jumlah objek, semakin lama waktu komputasinya.</p>		



Tabel 2.1. (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
5	Klasifikasi Citra Anjing Dan Kucing Menggunakan Metode <i>Convolutional Neural Network (CNN)</i>	Agung Slamet Riyadi, Ire Puspa Wardhani, Susi Widayati, Seminar Nasional Teknologi Informasi dan Komunikasi STI&K (SeNTIK), 2021	Untuk mendeteksi objek citra kucing dan anjing, serta membedakan model citra antara kedua objek citra.	<ol style="list-style-type: none"> <li>1. Aplikasi program klasifikasi berbasis <i>CNN</i> dengan hasil pengujian pada sistem dan aplikasi <i>CNN</i> digunakan untuk mengenali objek anjing dan kucing.</li> <li>2. Capaian akurasi dan presisi sistem klasifikasi terhadap penganalan anjing dan kucing yang didapat sebesar 84,09%</li> </ol>	Penambahan <i>dataset</i> yang digunakan untuk <i>training</i> dengan resolusi gambar yang bagus dapat membuat model yang dipakai lebih baik dan mengurangi <i>overfitting</i> .	Perbedaan penelitian sebelumnya menggunakan 2 citra sedangkan penulis menggunakan 1 citra dengan 5 kelas
6	<i>Classification of dog and cat images using the CNN method</i>	Teguh Adriyanto, Risky Aswi Ramadhani, Risa Helintar, Aidina Risktyawana. ILKOM Jurnal Ilmiah Vol. 14, No. 3, December 2022	Mengklasifikasikan citra anjing dan kucing menggunakan algoritma <i>CNN</i>	<i>CNN</i> belum bekerja secara maksimal. Hal ini terbukti setelah sistem dianalisis menggunakan <i>Confusions Matrix</i> dengan <i>presisi</i> 45%, <i>recall</i> 45%, dan <i>f1-score</i> 45%. Tingkat akurasi, <i>presisi</i> , <i>recall</i> dan <i>f1-score</i> kecil, oleh karena itu perlu adanya perbaikan pada sistem.	Hasil pada penelitian ini memiliki akurasi yang rendah, oleh karena itu memerlukan penelitian lebih lanjut agar memperoleh hasil akurasi yang lebih baik	Menggunakan algoritma <i>CNN</i> , yang membedakan penelitian sebelumnya menggunakan dua citra yaitu anjing dan kucing,

Tabel 2.1. (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
						sedangkan pada penelitian ini menggunakan 1 citra dengan 2 skenario
7	<i>Image Classification – Cat and Dog Images</i>	Tushar Jajodia, Pankaj Garg International Research Journal of Engineering and Technology (IRJET) e	Masalah utamanya adalah klasifikasi gambar, yang didefinisikan sebagai memprediksi kelas dari gambar tersebut	Dalam makalah ini, kami membangun sebuah jaringan saraf konvolusi dalam mendalam untuk klasifikasi gambar (gambar kucing dan anjing). Meskipun hanya menggunakan subset dari gambar-gambar tersebut, kami berhasil mencapai akurasi sebesar 90,10%. Jika seluruh dataset digunakan, akurasi akan menjadi lebih baik lagi.	Menggunakan lebih banyak dataset untuk percobaan dan testing	Perbedaan penelitian sebelumnya menggunakan 2 citra sedangkan penulis menggunakan 1 citra
8.	Deteksi Ras Kucing Menggunakan Compound Model Scaling CNN	Nadia Azahro Choirunisa, Tita Karlita, Rengga Asmara.	Mengklasifikasikan kucing berdasarkan ras menggunakan CNN	Penelitian menggunakan arsitektur EfficientNet-B0 sebagai base model	1. Menambahkan kelas pada dataset untuk jenis ras kucing	Pada penelitian terdahulu menggunakan 9

Tabel 2.1. (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
		Technomedia Journal (TMJ) p-ISSN: 2655-8807 Vol. 6 No. 2 Februari 2022		mendapatkan akurasi tertinggi yaitu 98% menggunakan optimizer Adam dan RMSprop dengan learning rate 0.001, masih terjadi overfitting pada model ini. Model paling optimal mendapatkan akurasi sebesar 95% dan akurasi validasi sebesar 91%. Model ini berhasil mengklasifikan 179 dari 180 gambar dengan kelas yang benar Model ini yaitu menggunakan optimizer RMSprop dengan learning rate 0.0001 dan menggunakan data pre-processing. Semakin kecil learning rate semakin kecil terjadinya overfitting, namun akurasi lebih rendah.	yang ciri-cirinya lumayan mirip. 2. Menambahkan variasi pada dataset seperti warna bulu, corak, atau ras campuran. Menggunakan metode pre-processing lainnya agar model lebih optimal.	kelas kucing, sedangkan penelitian ini menggunakan 5 kelas kucing dengan 2 skenario

Tabel 2.1. (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
9	Analisis Efek Augmentasi Dataset Dan Fine Tune Pada Algoritma Pre-Trained Convolutional Neural Network (CNN)	Theopilus Bayu Sasongko, Haryoko, Agit Amrullah, Jurnal Teknologi Informasi dan Ilmu Komputer (JTIK), 2023	Mengukur pengaruh teknik fine tune dan augmentasi dataset pada model transfer learning CNN mobilenet, efficientnet, dan nasnetmobile dengan dataset yang sangat kecil.	<ol style="list-style-type: none"> <li>1. Penelitian mengenai pengukuran pengaruh teknik fine tune &amp; augmentasi dataset pada model transfer learning CNN mobilenet, efficientnet, dan nasnetmobile dengan dataset yang kecil.</li> <li>2. Model efisien transfer learning (mobilenet, efficientnet, dan nasnetmobile), Teknik augmentasi random erase dan zoom range mendominasi peningkatan akurasi</li> <li>3. Peningkatan akurasi setelah augmentasi random erase ataupun zoom range yang terjadi sekitar 0.03% - 0.1%.</li> </ol>	<ol style="list-style-type: none"> <li>1. Pengujian lebih banyak metode augmentasi dataset</li> <li>2. Otomatisasi pencarian range nilai pada metode augmentasi</li> <li>3. Melakukan otomatisasi pemilihan teknik augmentasi dataset</li> <li>4. Untuk mengetahui kehandalan model maka pada penelitian berikutnya perlu dilakukan justifikasi model dengan dataset yang lainnya.</li> </ol>	Pada penelitian terdahulu menggunakan Teknik fine tuning, sedangkan pada penelitian yang dilakukan menggunakan model konvolusi (layer conv 2d) , lapisan pooling, lapisan dropout, lapisan flatten, dan lapisan dense

## 2.3. Landasan Teori

### 2.3.1. Kucing

Kucing adalah salah satu binatang kesayangan yang banyak dijadikan hewan peliharaan yang populer. Hal ini karena kucing mempunyai keistimewaan tersendiri yaitu warna bulu, bentuk tubuh, dan warna mata. Dalam bahasa latin kucing disebut dengan *Fell silveltris catus* yaitu sejenis binatang karnivora. Banyaknya jumlah ras kucing yang diakui internasional seperti Anggora, Abyssinian, Bengal, Birman, Persia dan Sphynx. Kucing persia memiliki ciri hidung pesek, tubuh agak membulat, dan berbulu panjang diseluruh tubuhnya. Sedangkan kucing Anggora biasanya berhidung mancung, berbadan agak ramping dan bulunya Panjang pada bagian-bagian tertentu saja. (Tita Rahayu, 2015). Tabel 2.2. menunjukkan taksonomi kucing dengan nama ilmiah *Fellis Catus*.

Tabel 2.2. Taksonomi Kucing

1	Kingdom	Animalia
2	Filum	Chordata
3	Subfilum	Vertebrata
4	Kelas	Mamalia
5	Ordo	Carnivora
6	Famili	Felidea
7	Genus	Felis
8	Spesies	Felis Catus

Secara umum kucing mempunyai bentuk tubuh yang hampir sama antara ras yang satu dengan yang lain baik itu ras, kucing local, ataupun kucing liar.



Claudio Ottoni dari University of Rome dalam penelitiannya mengatakan domestikasi kucing terjadi dalam dua jenis, kucing domestic mempunyai nenek moyang yang sama yaitu berasal dari kucing liar Arika Utara atau Asia Barat. Selain itu dengan mempelajari DNA kucing purba dari seluruh dunia, para peneliti menemukan domestikasi kucing terjadi pada periode Neolitik dan kemudian sampai pad Mesir kuno. Claudio Ottoni juga menyampaikan mengapa bias menyebar di seluruh dunia. Dengan menganalisis penemuan DNA kucing di kota pelabuhan, para peneliti mengambil kesimpulan bahwa kucing dibawa oleh kapal kemudian dijadikan penjaga Gudang makanan dari ancaman hewan penerat. Hal inilah yang menyebabkan kucing dapat menyebar di seluruh dunia.

Penelitian yang dilakukan oleh R. Ngitung (2021) mengenai karakteristik perilaku kucing menjelaskan bahwa sebagai hewan domestic, perilaku kucing dapat dikelompokkan kedaam 10 pola perilaku yaitu perilaku ingestif (perilaku makan), perilaku eliminative (perilaku membuang kotoran), perilaku seks (perilaku kawin), perilaku investigative (mengendus benda yang ada disekitarnya), perilaku agonistic (perilaku menantang), perilaku allelomimetic (perilaku meniru), perilaku mencari perlindungan, perilaku epimeletic (perilaku induk kucing yang menyayangi anaknya), perilaku et-epimeletic (hubungan era tantara kucing dengan pemiliknya) dan perilaku maladaptive (suka bermain-main).

### **2.3.2. Machine learning**

*Machine learning* merupakan ilmu pengembangan algoritma dan model secara statistik yang digunakan sistem komputer untuk menjalan tugas tanpa instruksi eksplisit, yang mengandalkan pola dan inferensi sebagai gantinya.

*Machine learning* dapat berperan dalam pengendalian kualitas dan penelitian yang inovatif. *Machine learning* sendiri sering dikategorikan menjadi dua jenis yaitu *supervised* dan *unsupervised* seperti yang ditunjukkan pada Gambar 2.1.

Gagasan utama di balik *machine learning* adalah hubungan matematis yang ada antara semua kombinasi data input dan output. Model *machine learning* tidak mengetahui hubungan ini sebelumnya, tetapi dapat menerka jika diberikan set data yang cukup. Hal ini berarti setiap algoritme *machine learning* dibangun di seputar fungsi matematika yang dapat dimodifikasi. Prinsip yang mendasarinya dapat dipahami seperti ini:

1. Kami "melatih" algoritme tersebut dengan memberinya kombinasi *input/output* ( $i,o$ ) berikut – (2,10), (5,19), dan (9,31)
2. Algoritme tersebut mengkomputasi hubungan antara input dan output menjadi:  
$$o=3*i+4$$
3. Selanjutnya, kami memberinya input 7 dan memintanya untuk memprediksi output. Algoritme tersebut dapat secara otomatis menentukan output-nya menjadi 25.

Meskipun ini adalah pemahaman dasar, *machine learning* berfokus pada prinsip bahwa semua poin data kompleks dapat dihubungkan secara matematis oleh sistem komputer selama sistem komputer tersebut memiliki data dan daya komputasi yang cukup untuk memproses data tersebut. Oleh karena itu, keakuratan output tersebut secara langsung berkorelasi dengan besarnya input yang diberikan.



Menurut (Mohri et.al, 2012) *machine learning* didefinisikan sebagai metode komputasi berdasarkan pengalaman untuk meningkatkan prediksi dengan tingkat akurasi yang tinggi. Terdapat dua skenario dalam *machie learning*, yaitu:

1. *Supervised learning*

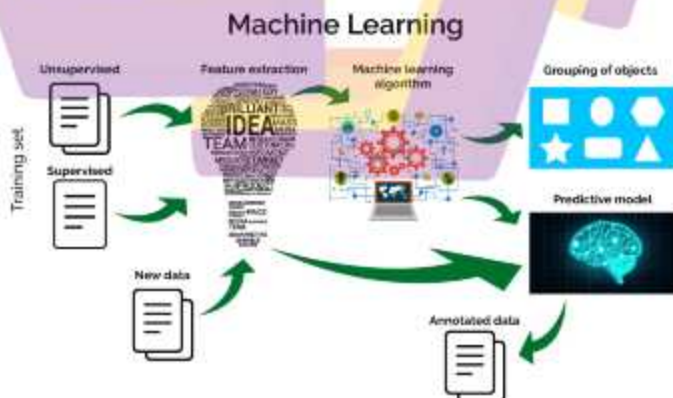
Merupakan pendekatan dalam *Artifial Intelligence* (AI) yang dilatih untuk mengenali pola antara input data dan label output.

2. *Unsepervised learning*

Merupakan teknik yang digunakan *machine learning* dalam pembuatan *artifial intelligence* yang dirancang untuk dapat “belajar sendiri” dalam mengumpulkan informasi, termasuk mengenali data yang tidak berlabel

3. *Reinforcement learning*

Merupakan fase pembelajaran dan tes saling yang digabung. Informasi diperoleh pembelajar yang secara aktif dengan berkomunikasi ke lingkungan sehingga mendapatkan unpan balik untuk setiap aksi dari pembelajar.



Gambar 2.1. Machine Learning (Pantech, 2018)

### 2.3.3. Data Mining

*Data mining* adalah proses untuk mendapatkan informasi yang berguna dari basis data yang besar dan perlu diekstraksi agar menjadi informasi baru dan dapat membantu dalam pengambilan keputusan (Suntoro, 2019:2). *Data mining* merupakan bidang ilmu yang digunakan untuk menangani masalah pengambilan informasi dari *batabase* yang besar dengan menggabungkan Teknik dan statistic, pembelajaran mesin, visualisasi data, pengenalan pola, dan database (Werdiningsih, 2020:17).

Tujuan dari *data mining* menurut Hoffer dkk, 2007 adalah:

1. *Explanatory*

Untuk menjelaskan beberapa kondisi penelitian, seperti mengapa penjualan truk *pick-up* meningkat di Colorado.

2. *Confirmatory*

Untuk mempertegas hipotesis, seperti hanya dua kali pendapatan keluarga lebih suka dipakai untuk membeli peralatan keluarga dibandingkan dengan satu kali pendapatan keluarga.

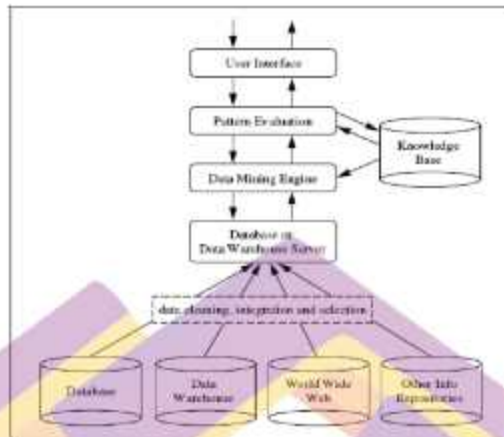
3. *Exploratory*

Untuk menganalisa data yang memiliki hubungan yang baru. misalnya, pola apa yang cocok untuk kasus penggelapan kartu kredit.

Menurut Werdiningsih, 2020:17 tujuan *data mining* adalah untuk mengekstrak informasi dengan metode cerdas dari kumpulan data kemudian mengubah informasi menjadi struktur yang dapat dipahami untuk penggunaan lebih lanjut.

Menurut Han dan Kamber, 2006 arsitektur system *data mining* memiliki komponen-komponen utama, yaitu:

1. *Database, data warehouse, world wide web*, atau tempat penyimpanan informasi lainnya: bisa berbentuk satu atau banyak database, *data warehouse, spreadsheet*, ataupun tempat penyimpanan informasi lainnya. *data cleaning, data integration* dan *data selection* dapat dijalankan pada data tersebut.
2. *Database dan data warehouse server*. Komponen ini bertanggungjawab dalam pengambilan data yang relevan, berdasarkan permintaan pengguna.
3. *Knowledge Based*. Komponen ini merupakan domain *knowledge* yang digunakan untuk memandu pencarian atau mengevaluasi pola-pola yang dihasilkan. Pengetahuan tersebut meliputi hirarki konsep yang digunakan untuk mengorganisasikan atribut atau nilai atribut kedalam level abstraksi yang berbeda. Pengetahuan tersebut juga dapat berupa kepercayaan pengguna (*user belief*), yang dapat digunakan untuk menentukan kemenarikan pola yang diperoleh.
4. *Data mining engine*. Bagian ini merupakan komponen penting dalam arsitektur system *data mining*. Komponen ini terdiri dari modul-modul fungsional seperti karakterisasi, asosiasi, klasifikasi, dan analisis *cluster*.
5. *Graphical User Interface (GUI)*. Modul ini berkomunikasi dengan pengguna dan *data mining*, melalui komponen ini, pengguna berinteraksi dengan system menggunakan *query*.



Gambar 2.2. Arsitektur Sistem Data mining (Han dan Kamber, 2006)

#### 2.3.4. Deep Learning

*Deep learning* adalah algoritma *neural network* yang menggunakan *metadata* sebagai input dan mengolah input tersebut menggunakan sekumpulan fungsi transformasi *non-linier* yang disusun mendalam dan berlapis-lapis. Algoritme *Deep Learning* menganalisis data dengan struktur logika yang mirip dengan yang digunakan oleh manusia. *Deep Learning* mempunyai kemampuan untuk mengolah fitur yang sesuai dari sebuah data untuk pemecahan masalah. *Deep learning* dengan *convolutional neural network* (CNN) banyak digunakan untuk melakukan klasifikasi, deteksi, dan prediksi pada gambar. Teori dari penelitian ini di ambil dari Review (LeCun, Bengio, & Hinton, 2015) yang berjudul *Deep learning* adalah aplikasi dari *machine learning* yang digunakan untuk mengidentifikasi objek dan gambar, menyalin pidato menjadi teks, mencocokkan item berita, dan menganalisis sentiment pengguna dari postingan di media sosial yang memanfaatkan kelas teknis. Metode *deep learning* merupakan metode *representation-learning* yang

terdiri dari berbagai tingkat representasi pada satu tingkat tertentu menjadi representasi pada tingkat yang lebih tinggi, ke tingkat yang abstrak.

### 2.3.5. *Algoritma*

Pengertian algoritma menurut para ahli adalah:

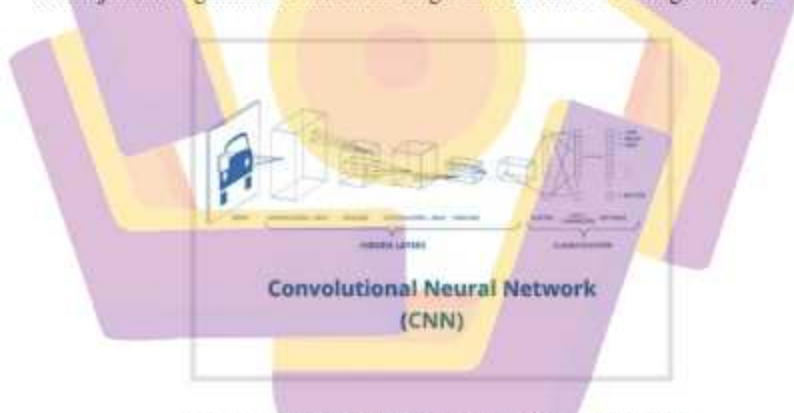
1. Algoritma menurut (Kai, 2020, 1.19) adalah suatu upaya dengan urutan operasi yang disusun secara logis dan sistematis untuk menyelesaikan suatu masalah untuk menghasilkan suatu output tertentu.
2. Algoritma berasal dari kata *algorism* dan *rithmis* yang pertama kali diperkenalkan oleh Abu Ja'far Muhammad Ibn Musa Khwarizmi pada 825 M di dalam buku *Al Jabr Wa-al Muqabla*. Dalam bidang pemrograman, algoritma didefinisikan sebagai metode yang terdiri dari serangkaian langkah yang terstruktur dan sistematis untuk menyelesaikan masalah dengan bantuan computer (Jano & Nani, 2018,5).
3. Algoritma menurut (Munir & Lidya, 2016,5) adalah urutan langkah-langkah untuk menyelesaikan suatu persoalan.
4. algoritma merupakan sekumpulan instruksi atau langkah-langkah yang dituliskan secara sistematis dan digunakan untuk penyelesaian masalah/persoalan logika dan matematika dengan bantuan computer (Sismoro, 2005,29)

### 2.3.6. *Convolutional neural network (CNN)*

*Deep learning* dengan *Convolutional Neural Network (CNN)* yang banyak digunakan untuk melakukan deteksi, klasifikasi, dan prediksi pada gambar.



Algoritma CNN memiliki *neuron* yang didesain untuk bekerja seperti lobus frontal, terutama pada bagian *visual cortex* pada manusia dan hewan. *Convolutional Neural Network* (CNN) adalah jenis jaringan syaraf tiruan khusus untuk memproses data yang dikenal memiliki *grid-like topologi*. (sindagi & Patel, 18) keberhasilan CNN dalam *computer visioan* telah mengilhami para peneliti untuk mengeksploitasi kemampuan mereka untuk mempelajari fungsi-fungsi *non-linier* dari *crowd images* yang sesuai dengan kerapatan peta atau jumlah yang sesuai. Berbagai metode berbasis CNN telah diusulkan dalam literatur. Kata *convolution* pada CNN yaitu sebuah operasi matematika pada dua buah fungsi yang kemudian menghasilkan fungsi ketiga. Fungsi ini menggabungkan 2 buah himpunan informasi dan menunjukkan bagaimana bentuk satu fungsi dimodifikasi oleh fungsi lainnya.



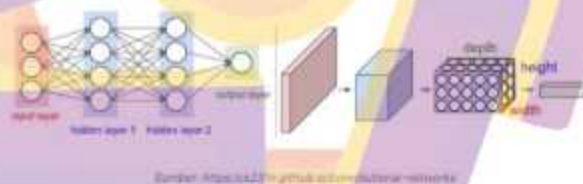
Gambar 2.3. Convolutional Neural Neteork (CNN)

CNN menggunakan *convolution* sebagai pengganti perkalian matriks umum. Operasi ini digunakan paling tidak pada satu lapisannya.



*Neural network* yang umum biasanya mengubah input dengan meletakkannya melalui rangkaian *hidden layer*, hal tersebut ditunjukkan pada Gambar 2.3. Setiap layer terdiri dari sekumpulan *neuron*, dimana setiap layer terhubung secara penuh dengan semua neuron pada layer sebelumnya. Terakhir, lapisan yang sudah terhubung sepenuhnya (*output layer*) digunakan untuk mewakili prediksi. Tidak seperti *neural network* biasa, lapisan pada algoritma CNN yang ditunjukkan pada Gambar 2.4 memiliki neuron yang diatur dalam 3 dimensi: *width*, *height*, dan *depth*. Dimensi *depth* mengacu pada dimensi ketiga dari fungsi aktivasi, bukan kedalaman *neural network* atau jumlah total layer dalam jaringan.

Neuron-neuron dalam satu layer tidak terhubung ke semua neuron di layer berikutnya tetapi hanya ke sebagian kecil saja. Terakhir, hasil akhir akan direduksi menjadi satu vektor skor probabilitas, yang diatur sepanjang dimensi *depth*. Perbedaan arsitektur ini dapat dilihat pada gambar berikut:



Gambar 2.4. Lapisan Pada Algoritma CNN

Di sebelah kiri adalah visualisasi layer yang ada pada *neural network* biasa. Sedangkan sebelah kanan adalah visualisasi layer yang ada di CNN.

Dapat dilihat bahwa CNN memiliki dimensi *depth* yang membuatnya berbentuk 3D. Operasi *convolution* merupakan bagian penting dari CNN. Pada algoritma CNN, *convolution* dijalankan pada data input menggunakan sebuah filter atau kernel yang kemudian digunakan untuk memetakan fitur.

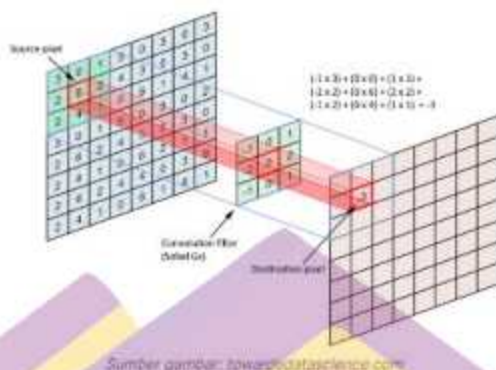
Operasi *convolution* dilakukan dengan menggeser filter di atas input. Di setiap lokasi, perkalian matriks dilakukan dan menjumlahkan hasilnya ke dalam peta fitur.

Gambar 2.5 di bawah menunjukkan cara kerja operasi *convolution*. Filter ditunjukkan dengan blok berwarna hijau bergerak di atas input (ditandai kotak berwarna biru) dan jumlah dari hasil operasi *convolution* masuk ke peta fitur (kotak merah)..



Gambar 2.5. Ilustrasi filter pada algoritma CNN

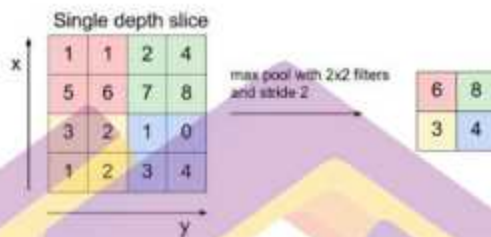
Untuk memudahkan penjelasan, visualisasi di atas berbentuk 2 dimensi. Tetapi yang sebenarnya terjadi adalah CNN berjalan dalam 3 dimensi ditunjukkan pada Gambar 2.6. Setiap gambar direpresentasikan sebagai matriks 3D dengan dimensi *width*, *height*, dan *depth*. *Depth* merupakan dimensi karena adanya warna yang digunakan dalam gambar (RGB).



Gambar 2.6. Ilustrasi 3D Convolution layer CNN

Beberapa operasi *convolution* pada data input akan dilakukan menggunakan filter yang berbeda. Hasilnya juga akan disimpan dalam peta fitur yang berbeda. Namun pada akhirnya, semua peta fitur ini akan diambil dan digabungkan sebagai hasil akhir dari *convolution* layer. Sama seperti *Neural Network* lainnya, kita menggunakan fungsi aktivasi (*activation function*) untuk membuat output kita menjadi *non-linear*. Dalam kasus *Convolutional Neural Network*, output dari operasi *convolution* akan dilewatkan melalui fungsi aktivasi. Salah satunya menggunakan fungsi aktivasi *ReLU*. Setelah *convolution* layer, biasanya ditambahkan lapisan penyatuan (*pooling layer*) di antara layer CNN. Fungsi pooling adalah untuk mengurangi dimensi secara terus menerus serta mengurangi jumlah parameter dan komputasi dalam jaringan. Hal ini akan mempersingkat waktu training dan mengontrol terjadinya *overfitting*. Jenis pooling yang paling sering digunakan pada CNN adalah *max pooling*. Pooling ini mengambil nilai maksimum di setiap *window/blok*. Ukuran *window/blok* ini perlu ditentukan

sebelumnya. Operasi ini mengurangi ukuran peta fitur dan hanya akan menyimpan informasi penting dan signifikan dalam proses klasifikasi yang ditunjukkan pada Gambar 2.7

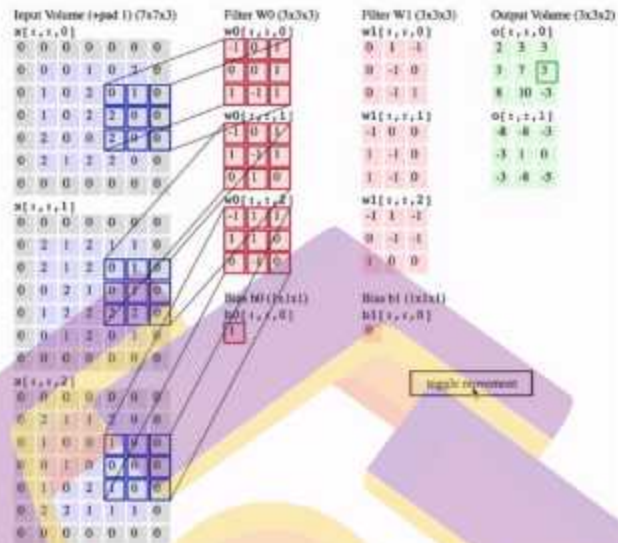


Gambar 2.7. Proses Klasifikasi

Ketika menggunakan CNN, ada 4 *hyperparameter* penting yang perlu kita tentukan, yakni:

1. Ukuran *kernel/filter*, yaitu panjang dan lebar filter yang akan digunakan. Misalnya 3x3
2. Jumlah *filter*, yaitu seberapa banyak *filter* yang akan kita gunakan
3. *Stride*, yaitu seberapa jauh step dari filter ketika digeser
4. *Padding*

Visualisasi dari cara kerja *convolution layer* pada CNN dapat dilihat pada Gambar 2.8 berikut ini:



Gambar 2.8. Visualisasi Cara Kerja CNN

Setelah melalui *convolution layer* dan *pooling layer*, komponen selanjutnya adalah klasifikasi yang terdiri dari beberapa *fully-connected layer* (layer yang terhubung secara penuh). Layer ini hanya dapat menerima data berdimensi 1.

Untuk mengonversi data 3 dimensi menjadi 1 dimensi, kita bisa menggunakan fungsi *flatten*.

*Neuron* pada *Fully-Connected (FC)* memiliki koneksi menyeluruh ke semua aktivasi di layer sebelumnya. Bagian ini pada prinsipnya sama dengan *Neural Network* biasa.

### 2.3.7. Confusion Matrix

*Confusion matrix* adalah salah satu alat ukur yang digunakan untuk menghitung tingkat akurasi dalam proses klasifikasi. Gambar 2.9 merupakan



gambaran untuk mempermudah pemahaman tentang istilah *confusion matrix* dalam klasifikasi.

		True Class	
		Positive	Negative
Predicted Class	Positive	TP	FP
	Negative	FN	TN

Gambar. 2.9. Confusion Matrix

*Actual values* merupakan nilai asli dari label atau class. *Predicted value* merupakan nilai prediksi hasil pemodelan mesin learning. *TP* adalah *True positive*; *FP* adalah *False positive*; *FN* adalah *False Negative* dan *TN* adalah *True Negative*.

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

Persamaan (1) adalah perhitungan nilai rata-rata nilai akurasi untuk menunjukkan tingkat efektifitas perkelas dari sebuah klasifikasi sokolova dan Lapalme (2009).

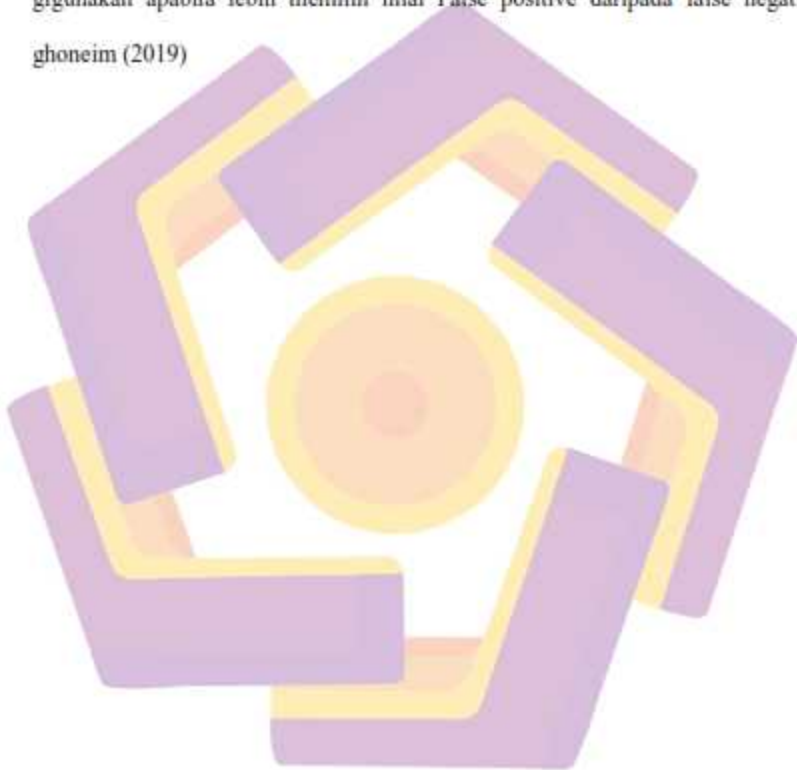
$$precision = \frac{TP}{TP + FP} \quad (2)$$

Persamaan (2) adalah perhitungan nilai *precision* yaitu dari data hasil klasifikasi seberapa banyak data yang benar antara nilai sebenarnya dengan prediksi yang diberikan oleh sistem.



$$recall = \frac{TP}{TP + FN} \quad (3)$$

Persamaan (3) adalah perhitungan nilai *recall* yaitu dari seluruh data benar dan seberapa banyak data yang keluar dalam hasil klasifikasi. Evaluasi *recall* digunakan apabila lebih memilih nilai False positive daripada false negative ghoneim (2019)



## BAB III

### METODE PENELITIAN

#### 3.1. Jenis, Sifat, dan Pendekatan Penelitian

Jenis dalam penelitian *eksperimental*. Penelitian *eksperimental* merupakan penelitian yang dilakukan untuk mengetahui pengaruh pemberian suatu *treatment* atau perlakuan terhadap subjek penelitian (Kasi, 2012). Penelitian ini bersifat kausal, yaitu mendapatkan informasi yang memiliki hubungan sebab akibat dengan penelitian kuantitatif yaitu menjelaskan bagaimana sesuatu dibangun dan dapat bekerja menggunakan skala numerik. Penelitian ini menggunakan *Convolutional Neural Network (CNN)* sebagai algoritma penyelesaian masalah berdasarkan citra. Percobaan yang dilakukan dalam penelitian ini adalah dengan melakukan pembelajaran terhadap klasifikasi citra kucing kemudian akan dievaluasi hasil pembelajaran tersebut untuk mengukur keakuratan dalam klasifikasi citra kucing dari setiap skenario untuk diketahui fakta-fakta dari hasil penelitian.

#### 3.2. Metode Pengumpulan Data

Pada penelitian ini dilakukan studi literasi tentang ras kucing, mengubah data dalam bentuk gambar, mempelajari struktur dari algoritma CNN melalui literasi jurnal, tutorial serta dokumentasi lainnya yang diperoleh dari *world wide web*. Data gambar tersebut diperoleh dari Oxford-IIIT. Proses pengumpulan *dataset* ditunjukkan pada gambar 3.1.



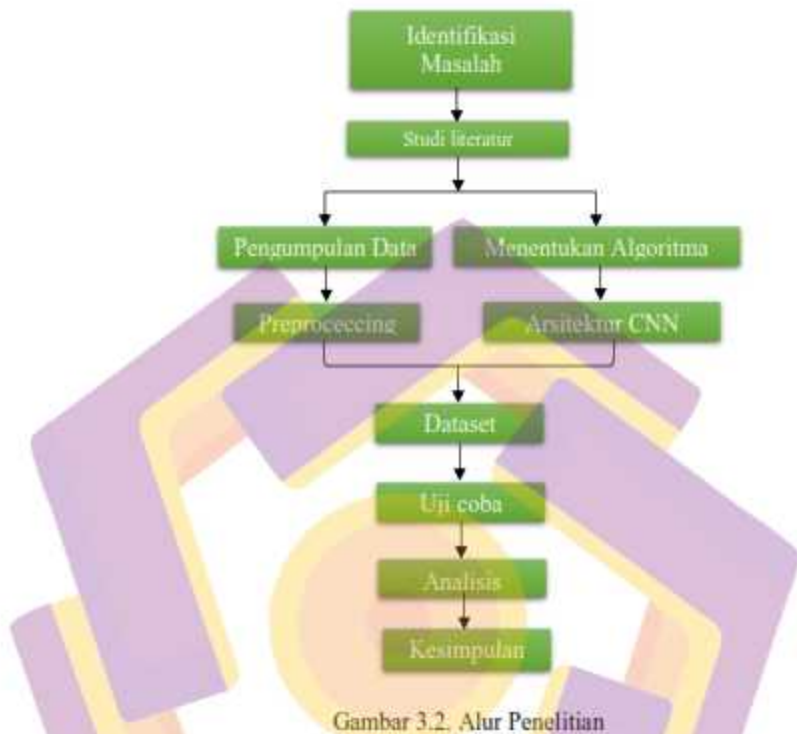
Gambar 3.1. Pengumpulan Data

### 3.3. Metode Analisis Data

Dalam penelitian ini, metode analisis data yang digunakan adalah analisis kuantitatif dengan algoritma *Convolutional Neural Network*. Tahap berikutnya *preprocessing Dataset* yang telah dikumpulkan berupa citra kucing yaitu dengan klasifikasi citra dan menyamakan ukuran citra, setelah *preprocessing* maka akan dilakukan proses *training* dengan menggunakan kombinasi dari arsitektur CNN. Kemudian akan diuji tingkat akurasi, presisi, recall dan *F-score* untuk mengetahui arsitektur mana yang paling baik tingkat akurasinya. Untuk melakukan penelitian ini akan dilakukan dengan menggunakan *google colab* dalam membangun arsitektur CNN.

### 3.4. Alur Penelitian

Alur dari penelitian ini meliputi identifikasi masalah, studi literatur, pengumpulan data, menentukan algoritma, preprocessing, uji coba, klasifikasi, analisis, kesimpulan. Tahap-tahap tersebut dirancang seperti pada gambar 3.2.



Untuk lebih jelasnya akan dijelaskan berikut ini:

1. Identifikasi Masalah

Proses indentifikasi masalah dalam penelitian ini dilakukan dengan membaca beberapa penelitian sebelumnya dan mencari tau permasalahan yang ada pada objek penelitian.

2. Studi literatur

Studi literatur dalam penelitian ini dilakukan dengna mencari dan membaca jurnal hasil penelitian dan buku yang sesuai dengan permasalahan

yang di angkat dalam penelitian ini sebagai bahan rujukan dalam menentukan metode dan algoritma yang sesuai dengna objek penelitian.

### 3. Pengumpulan Data

Proses Pengumpulan data citra kucing dan menentukan algoritma yang akan digunakan dalam penelitian yaitu *Convolutional Neural Network* (CNN). Setelah mencari dan membaca beberapa hasil penelitian sebagai rujukan, tahap berikutnya adalah tahap pengumpulan data berupa citra kucing yang di ambil dari internet.

### 4. Preprocessing

Tahap *preprocessing* dilakukan sebelum klasifikasi pada citra objek di laksanakan. Tahap-tahap tersebut antara lain: memberi label, menyamakan ukuran pixel pada semua citra

### 5. Dataset

Tahap preprocessing akan menghasilkan satu data set citra kucing yang terdiri dari 5 kelas (Abbyssian, Bengal, Birman, Persia, Sphynx)

### 6. Uji coba

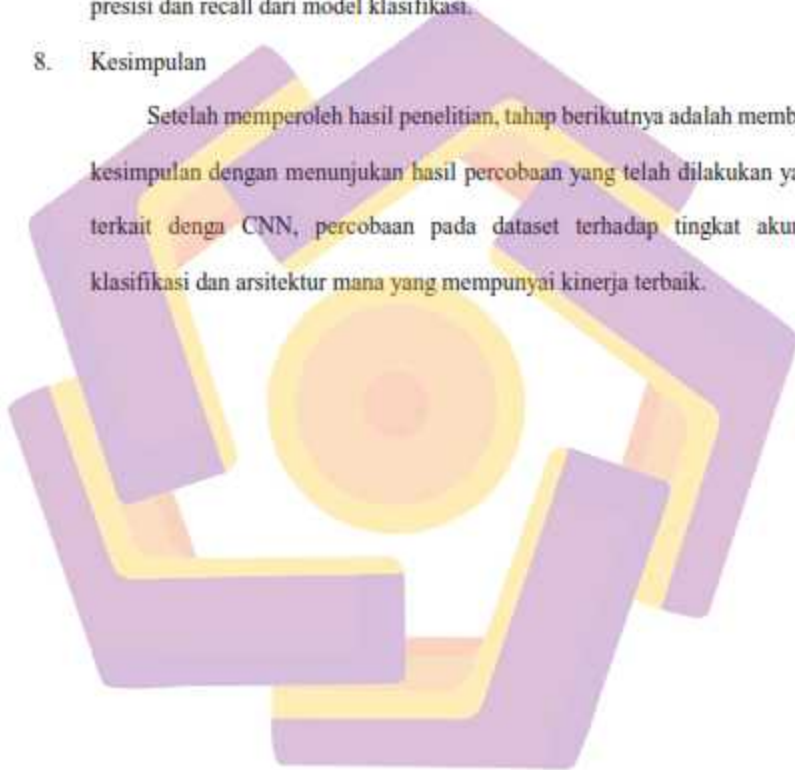
Tahap selanjutnya adalah uji coba menggunakan dataset citra kucing. Training data dilakukan dengan menggunakan *Convolutional Neural Network* (CNN) akan melalui proses *features extraction* oleh 2 lapisan dalam arsitektur CNN yaitu *convolutional layer* dan *pooling layer*. Proses ini berdasarkan citra kucing, langkah selanjutnya adalah proses uji coba yang terjadi pada lapisan *fully connected layer* hingga terbentuk model klasifikasi yang siap untuk di analisis.

## 7. Evaluasi hasil

Tahap evaluasi dilakukan dengan melakukan pengujian terhadap model klasifikasi yang telah dibuat dari sejumlah percobaan. Hasil yang diperoleh dalam percobaan ini adalah nilai-nilai *confusion matrix* yaitu nilai keakuratan, presisi dan recall dari model klasifikasi.

## 8. Kesimpulan

Setelah memperoleh hasil penelitian, tahap berikutnya adalah membuat kesimpulan dengan menunjukkan hasil percobaan yang telah dilakukan yang terkait dengan CNN, percobaan pada dataset terhadap tingkat akurasi klasifikasi dan arsitektur mana yang mempunyai kinerja terbaik.





## BAB IV

### HASIL PENELITIAN DAN PEMBAHASAN

Pada bab ini akan dijelaskan mengenai proses pengolahan citra kucing, hasil penelitian dan pembahasan klasifikasi citra kucing yang terdiri dari hasil preprocessing dan hasil training dan testing menggunakan CNN. Langkah-langkah yang dilakukan untuk mempersiapkan data dan proses pengolahan citra berdasarkan jenis-jenis citra kucing yang dibagi menjadi beberapa tahap diantaranya:

#### 4.1. Pengumpulan Data

Penelitian ini menggunakan data dari *Kaggle* yang terdiri dari 5 kelas yaitu Abyssinian, Bengal, Birman, Persia dan Sphynx. Dataset citra kucing tersebut diperoleh dari *Oxford-IIIT* yang kemudian di simpan dalam dalam *google drive* karena pengolahan citra kucing menggunakan *google colab*. Proses pelabelan kelas dilakukan dengan memberikan nama jenis kucing pada folder tempat penyimpanan citra kucing. Dataset yang diujicobakan pada penelitian ini terdiri dari Dua Set data, Data Pertama Setiap kelas terdiri dari 100 citra kucing yang berbeda, dan Data Set Kedua, setiap kelas terdiri dari 200 citra kucing yang berbeda.



Gambar 4.1. Pelabelan dataset

Gambar 4.1 menunjukkan proses pelabelan dataset citra kucing. Proses pelabelan dilakukan dengan memberikan nama jenis kucing pada folder tempat penyimpanan citra kucing. Proses Pelabelan pada kedua dataset melalui proses yang sama sehingga dapat diujicobakan pada model yang sama.

Table 4.1. Tabel contoh citra kucing

No	Kelas	Citra
1	Abyssinian	
2	Bengal	
3	Birman	
4	Persian	

Tabel 4.1. (Lanjutan)

No	Kelas	Citra
5	Sphynx	

Pada tabel 4.1 menunjukkan contoh citra kucing dengan latar belakang yang berbeda-beda, oleh karena itu memerlukan tahap *preprocessing* untuk mengolah dataset citra kucing sebelum dilakukan proses pelatihan model. Jumlah citra kucing yang digunakan dalam penelitian ini adalah sebagai berikut :

Tabel 4.2. Skenario Perbandingan Dataset yang digunakan

Nama Dataset	Jumlah Data	Jumlah Data Tiap Kelas	Jumlah Kelas/ Label	Data training	Data testing
Dataset satu (Skenario satu)	500	100	5	80 %	20 %
Dataset Dua (Skenario Dua)	1000	200	5	80 %	20 %

Dataset disimpan di dalam google drive, karena proses pengolahan citra kucing menggunakan google collab. Gambar 4.2 menunjukkan tampilan awal untuk mengawali penelitian ini.



Gambar 4.2. Workbook Phyton

Berdasarkan gambar 4.2 maka penulisan script program python dapat kita mulai dengan mengupload dataset yang telah disiapkan. Langkah berikutnya yaitu dengan menambahkan beberapa library yang akan dibutuhkan dalam penelitian ini maka yang di import sebagai berikut:

```
#import the required libraries
import os
import zipfile
import tensorflow as tf
from tensorflow.keras.optimizers import RMSprop
from tensorflow.keras.preprocessing.image import ImageDataGenerator
import matplotlib.pyplot as plt
```

*Tensorflow* merupakan salah satu library yang dikembangkan oleh google yang digunakan untuk mengembangkan dan menerapkan Machine Learning serta algoritma lain yang memiliki melakukan berbagai operasi matematika. *Tensorflow* memiliki dua tindakan utama yaitu membangun grafik komputasi dalam tahap konstruksi dan menjalankan grafik komputasi dalam fase eksekusi.

*Keras* adalah interface library yang *machine learning* yang bersifat *open-source* yang dibentuk oleh Bahasa pemrograman *python* dan dikembangkan oleh

Google Engineer yaitu Francois Chollet yang bertujuan untuk menyederhanakan implementasi algoritma-algoritma *deep learning* di atas *Tensorflow*. *Keras* digunakan untuk mengurangi jumlah tindakan yang diperlukan dalam mengimplementasikan kode umum serta dapat menjelaskan kesalahan pengguna agar dapat menentukan tindaklanjutnya. Tools ini dapat digunakan sebagai API untuk *Tensorflow*, perangkat jaringan komputasi seperti CNTK dan Theano untuk merancang, mengevaluasi, menerapkan, dan memvisualisasikan model *deep learning*. Selain itu *Keras* juga bisa mendukung hampir semua permodelan *neural network* seperti *convolutional*, *pooling*, *recurrent*, *embedding*, dan sejenisnya

*RMSprop (Root Mean Square Propagation)* adalah salah satu optimizer yang mempertahankan rata-rata dari kuadrat gradient untuk setiap bobot. *RMSProp* yang dikembangkan agar hasilnya lebih baik dan tidak mengurangi *learning rate* untuk mencegah agar tidak terjadi konvergensi. Algoritma optimizer juga membagi laju *learning rate* dengan rata-rata nilai gradien kuadrat yang hilang secara eksponensial.

$$\begin{aligned}
 v_{dw} &= \beta \cdot v_{dw} + (1 - \beta) \cdot dw^2 \\
 v_{db} &= \beta \cdot v_{db} + (1 - \beta) \cdot db^2 \\
 W &= W - \alpha \cdot \frac{dw}{\sqrt{v_{dw} + \epsilon}} \\
 b &= b - \alpha \cdot \frac{db}{\sqrt{v_{db} + \epsilon}}
 \end{aligned}$$

Gambar 4.3. Rumus *RMSProp* Optimizer



*Imagedatagenerator* digunakan untuk memperoleh *input* dari data asli dan selanjutnya melakukan transformasi data secara acak dan memberikan resultan keluaran yang berisi data yang baru ditransformasikan.

Matplotlib.pyplot yaitu *interface* berbasis negara ke matplotlib yang memberikan cara membuat plot yang implisit yang bertindak sebagai mpengatur *GUI* citra.

#### 4.2. Pre Processing

Selanjutnya untuk menghubungkan data yang disimpan dalam google drive dengan google colabs, berikut script untuk menghubungkan keduanya.

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

Coding diatas adalah coding untuk memunculkan dan mengunduh data dari google drive. Hasil daripada coding tersebut adalah memanggil folder dalam dataset pada google drive, dan dengan pengolahan lebih lanjut dapat dilihat pada Gambar 4.4 dan 4.5 yang menunjukkan jumlah data yang terdapat pada masing-masing folder baik data latihan maupun validasi yaitu dengan perbandingan 80 data latihan dan 20 data validasi.

```
total Abyssinian in training: 80
total Bengal in training: 80
total Birman in training: 80
total Persian in training: 80
total Sphynx in training: 80
total Abyssinian in validation: 20
total Bengal in validation: 20
total Birman in validation: 20
total Persian in validation: 20
total Sphynx in validation: 20
```

Gambar 4.4. Jumlah data Latihan dan Validasi Dataset Satu

```

total Abyssinian in training: 160
total Bengal in training: 160
total Birman in training: 160
total Persian in training: 160
total Sphynx in training: 160
total Abyssinian in validation: 40
total Bengal in validation: 40
total Birman in validation: 40
total Persian in validation: 40
total Sphynx in validation: 40

```

Gambar 4.5. Jumlah data Latihan dan Validasi Dataset Dua

Gambar 4.5 adalah script untuk melihat file yang berada dalam folder yang berisi dataset dalam penelitian ini. Folder tersebut berisi dataset dengan nama-nama file sesuai jenis kucing dengan jumlah 500 file pada dataset satu dan 100 file pada dataset dua.

```

#!/et's see the files in the directories
train_ab_names = os.listdir(train_ab_dir)
print(train_ab_names[:10])
train_be_names = os.listdir(train_be_dir)
print(train_be_names[:10])
train_bi_names = os.listdir(train_bi_dir)
print(train_bi_names[:10])
train_pe_names = os.listdir(train_pe_dir)
print(train_pe_names[:10])
train_sp_names = os.listdir(train_sp_dir)
print(train_sp_names[:10])

```

Gambar 4.6. Menampilkan citra

Hasil dari script program tersebut di tampilkan pada Gambar 4.6

```

['Abyssinian_46.jpg', 'Abyssinian_73.jpg', 'Abyssinian_44.jpg',
['Bengal_92.jpg', 'Bengal_65.jpg', 'Bengal_42.jpg', 'Bengal_44
['Birman_28.jpg', 'Birman_84.jpg', 'Birman_34.jpg', 'Birman_58
['Persian_20 (1).jpg', 'Persian_16.jpg', 'Persian_25 (1).jpg',
['Sphynx_70.jpg', 'Sphynx_75.jpg', 'Sphynx_71.jpg', 'Sphynx_34
['Abyssinian_100.jpg', 'Abyssinian_91.jpg', 'Abyssinian_93.jpg
['Bengal_95.jpg', 'Bengal_91.jpg', 'Bengal_97.jpg', 'Bengal_89
['Birman_84.jpg', 'Birman_86.jpg', 'Birman_94.jpg', 'Birman_83
['Persian_112.jpg', 'Persian_105.jpg', 'Persian_114.jpg', 'Per:
['Sphynx_128.jpg', 'Sphynx_127.jpg', 'Sphynx_112.jpg', 'Sphynx

```



Gambar 4.7. Tampilan Citra Kucing

#### 4.3. Pembuatan Model Arsitektur CNN

Pembuatan model umumnya melibatkan proses konvolusi dengan fungsi aktivasi dan proses pooling. Jumlah proses ini disesuaikan dengan kebutuhan penelitian. Model ini dibangun dengan beberapa jenis lapisan, termasuk lapisan konvolusi (*layer\_conv\_2d*), lapisan pooling, lapisan dropout, lapisan flatten, dan lapisan dense.

Proses konvolusi dilakukan sebanyak 6 kali, ditunjukkan dengan jumlah convolution layer yang digunakan. Secara umum, 2 sampai 3 lapisan sudah cukup untuk mendapatkan model klasifikasi dengan akurasi tinggi. Penelitian ini menggunakan banyak lapisan untuk melatih model dan mengamati kinerjanya.

Convolution layer diaktifkan dengan menggunakan ReLU. ReLU atau *Rectified linier unit* merupakan salah satu fungsi aktivasi pada sebuah jaringan syaraf. Fungsi dari aktivasi ini digunakan pada *convolutional neural network*. ReLU

digunakan untuk menghilangkan sebuah nilai yang ada pada citra. Cara kerja ReLu yaitu dengan mengganti nilai negative yang ada dalam suatu citra menjadi bernilai 0. Jumlah filter yang digunakan dalam *convolution layer* yang pertama dan kedua adalah 32. Filter yang digunakan pada *convolution layer* ketiga dan keempat adalah 64. Pada layer kelima menggunakan 128, sedangkan layer ke enam menggunakan filter 256. Penggunaan jumlah filter yang lebih banyak pada dua lapisan konvolusi terakhir disebabkan oleh ukuran input pada kedua lapisan tersebut lebih kecil sehingga dibutuhkan lebih banyak filter untuk mengesktrak informasi citra.

Berikut adalah hasil pembuatan model arsitektur CNN pada dataset Satu dan Dua :

Tabel 4.3. Tabel hasil Model Dataset Satu

Lapisan (Type)	Bentuk Keluaran	Parameter
conv2d (Conv2D)	(None, 198, 198, 32)	896
max_pooling2d (MaxPooling2D)	(None, 99, 99, 32)	0
conv2d_1 (Conv2D)	(None, 97, 97, 32)	9248
max_pooling2d_1 (MaxPooling2D)	(None, 48, 48, 32)	0
batch_normalization (BatchNormalization)	(None, 48, 48, 32)	128
conv2d_2 (Conv2D)	(None, 46, 46, 64)	18496
max_pooling2d_2 (MaxPooling2D)	(None, 23, 23, 64)	0
conv2d_3 (Conv2D)	(None, 21, 21, 64)	36928
max_pooling2d_3 (MaxPooling2D)	(None, 10, 10, 64)	0
conv2d_4 (Conv2D)	(None, 8, 8, 128)	73856
max_pooling2d_4 (MaxPooling2D)	(None, 4, 4, 128)	0
conv2d_5 (Conv2D)	(None, 2, 2, 256)	295168
max_pooling2d_5 (MaxPooling2D)	(None, 1, 1, 256)	0
flatten (Flatten)	(None, 256)	0
dense (Dense)	(None, 800)	205600
dense_1 (Dense)	(None, 300)	240300
dense_2 (Dense)	(None, 5)	1505



Tabel 4.4. Tabel Hasil Model Dataset Dua

Lapisan (Tipe)	Bentuk Keluaran	Parameter
conv2d_12 (Conv2D)	(None, 198, 198, 32)	896
max_pooling2d_12 (MaxPooling2D)	(None, 99, 99, 32)	0
conv2d_13 (Conv2D)	(None, 97, 97, 32)	9248
max_pooling2d_13 (MaxPooling2D)	(None, 48, 48, 32)	0
batch_normalization_2 (BatchNormalization)	(None, 48, 48, 32)	128
conv2d_14 (Conv2D)	(None, 46, 46, 64)	18496
max_pooling2d_14 (MaxPooling2D)	(None, 23, 23, 64)	0
conv2d_15 (Conv2D)	(None, 21, 21, 64)	36928
max_pooling2d_15 (MaxPooling2D)	(None, 10, 10, 64)	0
conv2d_16 (Conv2D)	(None, 8, 8, 128)	73856
max_pooling2d_16 (MaxPooling2D)	(None, 4, 4, 128)	0
conv2d_17 (Conv2D)	(None, 2, 2, 256)	295168
max_pooling2d_17 (MaxPooling2D)	(None, 1, 1, 256)	0
flatten_2 (Flatten)	(None, 256)	0
dense_6 (Dense)	(None, 800)	205600
dense_7 (Dense)	(None, 300)	240300
dense_8 (Dense)	(None, 5)	1505

Setelah didapat model Analisa CNN kemudian dioptimasi dengan menggunakan RMS Prop dengan learning rate 0.001 seperti dalam script berikut :

```
model.compile(optimizer = RMSprop(learning_rate=0.001),
              loss = 'categorical_crossentropy', metrics=['accuracy'])
```

Gambar 4.7. Optimasi RMS Prop pada Python

Learning rate 0.001 dipilih dengan alasan sebagai berikut :

- o **Nilai umum:** 0.001 adalah nilai learning rate yang umum digunakan dalam berbagai model deep learning, termasuk CNN.



- o **Mempercepat konvergensi:** Nilai yang lebih kecil dapat membantu model mencapai konvergensi lebih cepat, terutama pada dataset yang besar.
- o **Mencegah overfitting:** Nilai yang lebih kecil dapat membantu mencegah overfitting, di mana model terlalu fokus pada data pelatihan dan tidak dapat digeneralisasikan dengan baik ke data baru.

#### 4.4. Analisis Data

Dataset yang telah berhasil diunduh dan dikelompokkan diberikan nama sesuai dengan nama jenis kucing, langkah berikutnya yaitu pengolahan dataset kucing agar dapat digunakan untuk *input* dalam klasifikasi citra kucing.

Pada penelitian ini dilakukan proses penggandaan data/citra pada data training melalui proses augmentasi. Berikut ini adalah bagian dari kode augmentasi citra.

```
train_datagen = ImageDataGenerator(  
    rescale=1./255.,  
    rotation_range=20,  
    width_shift_range=0.2,  
    height_shift_range=0.2,  
    shear_range=0.2,  
    zoom_range=0.2,  
    horizontal_flip=True,  
    fill_mode='nearest')
```

Gambar 4.8. Code Augmentasi

Bagian kode di atas merupakan list program untuk melakukan augmentasi citra pada data training. Pada penelitian ini *package ImageDataGenerator* dari *library Keras* digunakan untuk membantu proses *augmentasi*.

#### 4.5. Analisis Hasil Penelitian

Penelitian ini membagi data menjadi dua skenario yaitu dataset satu dan datadua seperti yang sudah dijelaskan pada sub bab sebelumnya.

Setelah melakukan proses pengolahan data, dan model dilatih sebanyak 180 epoch. Epoch merupakan satu rangkaian algoritma dari *mechine learning* 'latihan' dari dataset training yang akan diproses. Dalam *Convolutional Neural Network* proses latihan dilakukan secara berulang-ulang dengan tujuan untuk mencapai akurasi terbaik. Nilai dari epoch tidak diketahui secara pasti, karena itu jumlah data akan mempengaruhi jumlah pemakaian epoch. Dalam penelitian ini jumlah epoch yang digunakan sebanyak 180 dengan menggunakan dua scenario yang berbeda yaitu, dataset satu yang berjumlah 500 dan scenario dua dengan jumlah dataset 1000. Penggunaan dua scenario dilakukan untuk mengetahui perbandingan hasil akurasi dari masing-masing scenario dengan jumlah dataset yang berbeda tetapi menggunakan jumlah epoch yang sama. Hasil pengujian pada dua scenario dengan jumlah dataset yang berbeda dan jumlah epoch yang sama menghasilkan overffiting hal tersebut dikarenakan nilai akurasi pada proses traning lebih tinggi dari pada proses testing baik pada scenario satu maupun scenario dua. Jumlah epoch yang lebih tinggi menghasilkan jumlah akurasi yang lebih baik, hal ini didasarkan pada jurnal "Perbandingan Jumlah Epoch Dan Steps Per Epoch Pada Convolutional Neural Network Untuk Meningkatkan Akurasi Dalam Klasifikasi Gambar" oleh Enie Yuliani, dengan berlandaskan jurnal tersebut peneliti mengambil jumlah epoch yang cukup signifikan yaitu 180 epoch, angka ini didasarkan pada kemampuan komputasi yang digunakan oleh google colab. Setelah dilakukan

proses epoch didapatkan hasil untuk skenario dataset satu yang ditunjukkan pada tabel 4.5 sebagai berikut :

Tabel 4.5. Hasil Percobaan Skenario Pertama Dataset Satu

EPOCH	WAKTU	LOSS	ACCURACY	Val_LOSS	Val_ACCURACY
Epoch 1/180	142s 7s/step	loss: 1.6558	accuracy: 0.1875	val_loss: 1.6091	val_accuracy: 0.2000
Epoch 2/180	31s 2s/step	loss: 1.6144	accuracy: 0.2400	val_loss: 1.7461	val_accuracy: 0.2000
Epoch 3/180	30s 1s/step	loss: 1.5997	accuracy: 0.2725	val_loss: 1.6059	val_accuracy: 0.2100
Epoch 4/180	32s 2s/step	loss: 1.5421	accuracy: 0.3175	val_loss: 1.5913	val_accuracy: 0.2100
Epoch 5/180	29s 1s/step	loss: 1.4933	accuracy: 0.3400	val_loss: 1.6054	val_accuracy: 0.2000
Epoch 6/180	30s 2s/step	loss: 1.4590	accuracy: 0.3475	val_loss: 1.5496	val_accuracy: 0.2400
Epoch 7/180	30s 1s/step	loss: 1.5129	accuracy: 0.3400	val_loss: 1.6035	val_accuracy: 0.2100
Epoch 8/180	30s 1s/step	loss: 1.4497	accuracy: 0.3525	val_loss: 1.6143	val_accuracy: 0.2100
Epoch 9/180	35s 2s/step	loss: 1.4160	accuracy: 0.3800	val_loss: 1.5424	val_accuracy: 0.2300
Epoch 10/180	39s 2s/step	loss: 1.4126	accuracy: 0.3925	val_loss: 1.6183	val_accuracy: 0.2500

Tabel 4.5. (Lanjutan)

Epoch	34s	loss:	accuracy:	val_loss:	val_accuracy:
11/180	2s/step	1.3968	0.4250	1.9981	0.2000
Epoch	31s	loss:	accuracy:	val_loss:	val_accuracy:
12/180	2s/step	1.3522	0.3950	1.8935	0.2000
Epoch	31s	loss:	accuracy:	val_loss:	val_accuracy:
13/180	2s/step	1.3854	0.4000	1.4309	0.3800
Epoch	31s	loss:	accuracy:	val_loss:	val_accuracy:
14/180	2s/step	1.2733	0.4375	1.5564	0.3800
Epoch	31s	loss:	accuracy:	val_loss:	val_accuracy:
15/180	2s/step	1.2924	0.4600	1.4882	0.2900
Epoch	31s	loss:	accuracy:	val_loss:	val_accuracy:
16/180	2s/step	1.2701	0.4975	1.6035	0.3100
Epoch	30s	loss:	accuracy:	val_loss:	val_accuracy:
17/180	1s/step	1.1803	0.5075	1.7605	0.2900
Epoch	31s	loss:	accuracy:	val_loss:	val_accuracy:
18/180	2s/step	1.2060	0.5150	1.2255	0.5800
Epoch	33s	loss:	accuracy:	val_loss:	val_accuracy:
19/180	2s/step	1.2629	0.5000	1.2364	0.5100
Epoch	33s	loss:	accuracy:	val_loss:	val_accuracy:
20/180	2s/step	1.1945	0.5400	1.5769	0.3500
Epoch	30s	loss:	accuracy:	val_loss:	val_accuracy:
101/180	1s/step	0.4700	0.8400	0.5828	0.8200
Epoch	36s	loss:	accuracy:	val_loss:	val_accuracy:
102/180	2s/step	0.4738	0.8475	0.9711	0.7700

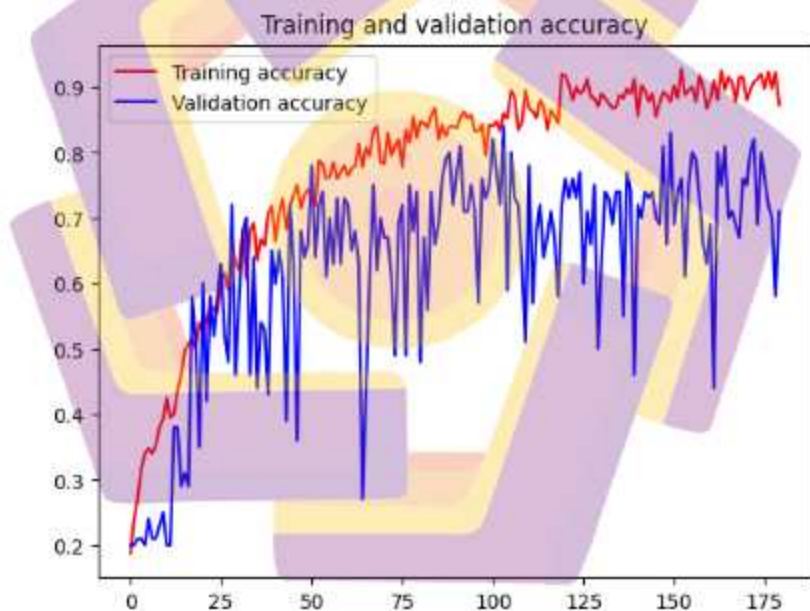
Tabel 4.5. (Lanjutan)

Epoch	31s	loss:	accuracy:	val_loss:	val_accuracy:
103/180	2s/step	0.4713	0.8325	1.0922	0.7200
Epoch	32s	loss:	accuracy:	val_loss:	val_accuracy:
104/180	2s/step	0.3769	0.8600	0.6870	0.8400
Epoch	30s	loss:	accuracy:	val_loss:	val_accuracy:
105/180	1s/step	0.4926	0.8525	1.2528	0.5900
.....	.....	.....	.....	.....	.....
Epoch	31s	loss:	accuracy:	val_loss:	val_accuracy:
150/180	2s/step	0.4268	0.8775	0.9111	0.8300
Epoch	31s	loss:	accuracy:	val_loss:	val_accuracy:
151/180	2s/step	0.3692	0.9025	1.3722	0.6900
Epoch	32s	loss:	accuracy:	val_loss:	val_accuracy:
152/180	2s/step	0.4470	0.8825	1.0678	0.7400
Epoch	30s	loss:	accuracy:	val_loss:	val_accuracy:
153/180	1s/step	0.2527	0.9275	1.2834	0.7600
Epoch	30s	loss:	accuracy:	val_loss:	val_accuracy:
154/180	2s/step	0.3786	0.8875	1.2868	0.6100
Epoch	31s	loss:	accuracy:	val_loss:	val_accuracy:
155/180	2s/step	0.3663	0.8950	1.0260	0.7500
.....	.....	.....	.....	.....	.....
Epoch	31s	loss:	accuracy:	val_loss:	val_accuracy:
176/180	2s/step	0.3179	0.8975	0.9764	0.7600
Epoch	31s	loss:	accuracy:	val_loss:	val_accuracy:
177/180	2s/step	0.1803	0.9225	1.5474	0.7200



Tabel 4.5. (Lanjutan)

Epoch	35s	loss:	accuracy:	val_loss:	val_accuracy:
178/180	2s/step	0.3722	0.8975	1.4913	0.7000
Epoch	30s	loss:	accuracy:	val_loss:	val_accuracy:
179/180	2s/step	0.2591	0.9225	5.5539	0.5800
Epoch	31s	loss:	accuracy:	val_loss:	val_accuracy:
180/180	2s/step	0.4872	0.8725	1.2173	0.7100



Gambar 4.9. Grafik Akurasi Training dan Validasi Dataset Satu

Gambar 4.9 menunjukkan hasil pengolahan pada dataset satu dengan akurasi tertinggi pada training adalah 92,75% dan loss pada 0.2527, sedangkan untuk data

validasi akurasi tertinggi adalah 84% dan loss pada 0.6870. Sedangkan untuk skenario Dataset dua disajikan pada tabel 4.6 berikut :

Tabel 4.6. Hasil Percobaan Skenario Pertama Dataset Dua

EPOCH	TIME	LOSS	ACCURACY	LOSS	ACCURACY
Epoch 1/180	122s 6s/step	loss: 1.6587	accuracy: 0.1940	val_loss: 1.6098	val_accuracy: 0.2800
Epoch 2/180	50s 3s/step	loss: 1.6122	accuracy: 0.2075	val_loss: 1.6087	val_accuracy: 0.1400
Epoch 3/180	41s 2s/step	loss: 1.6232	accuracy: 0.2443	val_loss: 1.6069	val_accuracy: 0.2400
Epoch 4/180	35s 2s/step	loss: 1.5866	accuracy: 0.3075	val_loss: 1.6167	val_accuracy: 0.1200
Epoch 5/180	32s 2s/step	loss: 1.5577	accuracy: 0.3199	val_loss: 1.6023	val_accuracy: 0.2700
Epoch 6/180	32s 2s/step	loss: 1.5303	accuracy: 0.3325	val_loss: 1.6113	val_accuracy: 0.2300
Epoch 7/180	32s 2s/step	loss: 1.5230	accuracy: 0.3476	val_loss: 1.5805	val_accuracy: 0.2400
Epoch 8/180	32s 2s/step	loss: 1.4678	accuracy: 0.3475	val_loss: 1.5818	val_accuracy: 0.1500
Epoch 9/180	33s 2s/step	loss: 1.4785	accuracy: 0.3854	val_loss: 1.5814	val_accuracy: 0.2500
Epoch 10/180	32s 2s/step	loss: 1.4109	accuracy: 0.4375	val_loss: 1.5274	val_accuracy: 0.3000
Epoch 11/180	32s 2s/step	loss: 1.4143	accuracy: 0.3879	val_loss: 1.5519	val_accuracy: 0.2600
Epoch 12/180	39s 2s/step	loss: 1.3126	accuracy: 0.4912	val_loss: 1.5350	val_accuracy: 0.2900
Epoch 13/180	32s 2s/step	loss: 1.3973	accuracy: 0.4125	val_loss: 1.4801	val_accuracy: 0.3600
Epoch 14/180	31s 2s/step	loss: 1.3056	accuracy: 0.4685	val_loss: 1.6206	val_accuracy: 0.3200
Epoch 15/180	34s 2s/step	loss: 1.3373	accuracy: 0.4525	val_loss: 1.4708	val_accuracy: 0.4500
Epoch 16/180	31s 2s/step	loss: 1.3745	accuracy: 0.4675	val_loss: 1.4126	val_accuracy: 0.4900
Epoch 17/180	31s 2s/step	loss: 1.1938	accuracy: 0.5466	val_loss: 1.4962	val_accuracy: 0.3400
Epoch 18/180	31s 2s/step	loss: 1.2672	accuracy: 0.4710	val_loss: 1.4488	val_accuracy: 0.4800

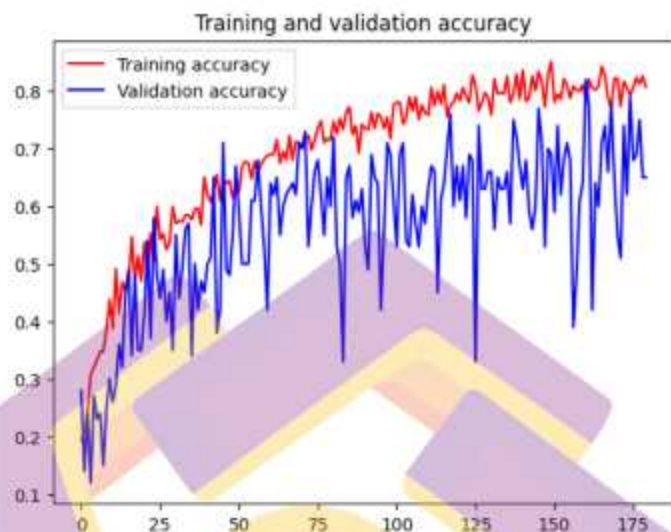
Tabel 4.6. (Lanjutan)

Epoch 20/180	32s 2s/step	loss: 1.3282	accuracy: 0.4700	val_loss: 1.4546	val_accuracy: 0.3500
Epoch 145/180	31s 2s/step	loss: 0.5462	accuracy: 0.8010	val_loss: 1.4674	val_accuracy: 0.6000
Epoch 146/180	33s 2s/step	loss: 0.4565	accuracy: 0.8338	val_loss: 0.8438	val_accuracy: 0.7700
Epoch 147/180	32s 2s/step	loss: 0.5264	accuracy: 0.8175	val_loss: 0.8608	val_accuracy: 0.7100
Epoch 148/180	31s 2s/step	loss: 0.6414	accuracy: 0.7725	val_loss: 1.3746	val_accuracy: 0.5300
Epoch 149/180	35s 2s/step	loss: 0.5924	accuracy: 0.8300	val_loss: 0.9698	val_accuracy: 0.7000
Epoch 150/180	31s 2s/step	loss: 0.5089	accuracy: 0.8514	val_loss: 1.1428	val_accuracy: 0.6900
Epoch 151/180	31s 2s/step	loss: 0.6552	accuracy: 0.7809	val_loss: 1.1479	val_accuracy: 0.5900
Epoch 152/180	31s 2s/step	loss: 0.5969	accuracy: 0.7950	val_loss: 0.7464	val_accuracy: 0.7400
Epoch 153/180	35s 2s/step	loss: 0.5883	accuracy: 0.7859	val_loss: 1.2468	val_accuracy: 0.6200
Epoch 154/180	31s 2s/step	loss: 0.6122	accuracy: 0.8212	val_loss: 1.0619	val_accuracy: 0.6800
Epoch 155/180	32s 2s/step	loss: 0.5336	accuracy: 0.8100	val_loss: 0.8712	val_accuracy: 0.7100
Epoch 156/180	31s 2s/step	loss: 0.5672	accuracy: 0.7875	val_loss: 1.4051	val_accuracy: 0.6800
Epoch 157/180	33s 2s/step	loss: 0.5120	accuracy: 0.8086	val_loss: 2.4918	val_accuracy: 0.3900
Epoch 158/180	31s 2s/step	loss: 0.7006	accuracy: 0.8025	val_loss: 1.9105	val_accuracy: 0.4800
Epoch 159/180	35s 2s/step	loss: 0.5691	accuracy: 0.8000	val_loss: 1.1280	val_accuracy: 0.6100
Epoch 160/180	36s 2s/step	loss: 0.5754	accuracy: 0.8111	val_loss: 1.2081	val_accuracy: 0.6400
Epoch 161/180	33s 2s/step	loss: 0.6769	accuracy: 0.7834	val_loss: 0.7747	val_accuracy: 0.8200
Epoch 162/180	33s 2s/step	loss: 0.5392	accuracy: 0.8200	val_loss: 0.6504	val_accuracy: 0.7700
Epoch 163/180	34s 2s/step	loss: 0.5207	accuracy: 0.8050	val_loss: 3.0289	val_accuracy: 0.4200
Epoch 164/180	36s 2s/step	loss: 0.6665	accuracy: 0.8025	val_loss: 0.9784	val_accuracy: 0.6400

Tabel 4.6. (Lanjutan)

Epoch 165/180	33s 2s/step	loss: 0.5999	accuracy: 0.8086	val_loss: 1.4078	val_accuracy: 0.6000
Epoch 166/180	33s 2s/step	loss: 0.4158	accuracy: 0.8425	val_loss: 1.6000	val_accuracy: 0.7100
Epoch 167/180	33s 2s/step	loss: 0.5264	accuracy: 0.8287	val_loss: 0.9429	val_accuracy: 0.7400
Epoch 168/180	33s 2s/step	loss: 0.5716	accuracy: 0.7875	val_loss: 0.9366	val_accuracy: 0.6600
Epoch 169/180	33s 2s/step	loss: 0.5370	accuracy: 0.7700	val_loss: 0.8362	val_accuracy: 0.7800
Epoch 170/180	35s 2s/step	loss: 0.5239	accuracy: 0.8225	val_loss: 1.3083	val_accuracy: 0.6500
Epoch 171/180	38s 2s/step	loss: 0.4529	accuracy: 0.8225	val_loss: 1.4726	val_accuracy: 0.5500
Epoch 172/180	30s 2s/step	loss: 0.7288	accuracy: 0.7733	val_loss: 1.4001	val_accuracy: 0.5100
Epoch 173/180	31s 2s/step	loss: 0.5756	accuracy: 0.8175	val_loss: 0.8258	val_accuracy: 0.7400
Epoch 174/180	32s 2s/step	loss: 0.4872	accuracy: 0.8060	val_loss: 1.6733	val_accuracy: 0.6200
Epoch 175/180	31s 2s/step	loss: 0.6967	accuracy: 0.7925	val_loss: 0.6410	val_accuracy: 0.7900
Epoch 176/180	33s 2s/step	loss: 0.5353	accuracy: 0.8060	val_loss: 1.2814	val_accuracy: 0.6800
Epoch 177/180	33s 2s/step	loss: 0.5801	accuracy: 0.8237	val_loss: 1.1358	val_accuracy: 0.6900
Epoch 178/180	32s 2s/step	loss: 0.4961	accuracy: 0.8125	val_loss: 0.7262	val_accuracy: 0.7500
Epoch 179/180	31s 2s/step	loss: 0.5389	accuracy: 0.8262	val_loss: 1.0307	val_accuracy: 0.6500
Epoch 180/180	30s 2s/step	loss: 0.5289	accuracy: 0.8075	val_loss: 1.0665	val_accuracy: 0.6500





Gambar 4.10. Grafik Akurasi Training dan Validasi Dua Dataset Dua

Hasil pengolahan data pada dataset satu ditunjukkan pada gambar 4.10 dengan akurasi tertinggi pada training adalah 85,14% dan loss pada 0.5089, sedangkan untuk data validasi akurasi tertinggi adalah 82% dan loss pada 0.7747.

Hasil pengolahan pada dataset satu dan dua menunjukkan bahwa kedua percobaan mengalami *overfitting*, yaitu dimana hasil akurasi pada proses training lebih tinggi dibandingkan dengan akurasi pada proses testing seperti yang ditunjukkan pada tabel 4.7 berikut:

Tabel 4.7. Perbandingan Akurasi Training dan Testing

Nama Dataset	Training		Testing	
	Akurasi	Loss	Akurasi	Loss
Dataset satu (Skenario satu)	92,75%	0.2527	84%	0.6870
Dataset Dua (Skenario Dua)	85,14%	0.5089	82%	0.7747



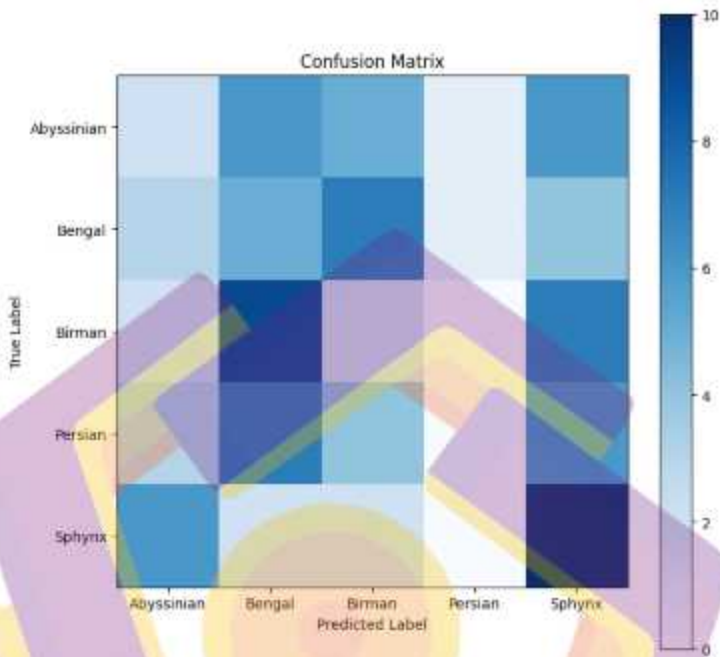
#### 4.6. Evaluasi Kinerja

Pada tahapan ini evaluasi ditunjukkan dengan menggunakan *confusion matrix* pada dataset satu dan dua, serta nilai precision dan recall seperti pada gambar dan tabel berikut ini :

Tabel 4.8. Precision, Recall dan F1 Score

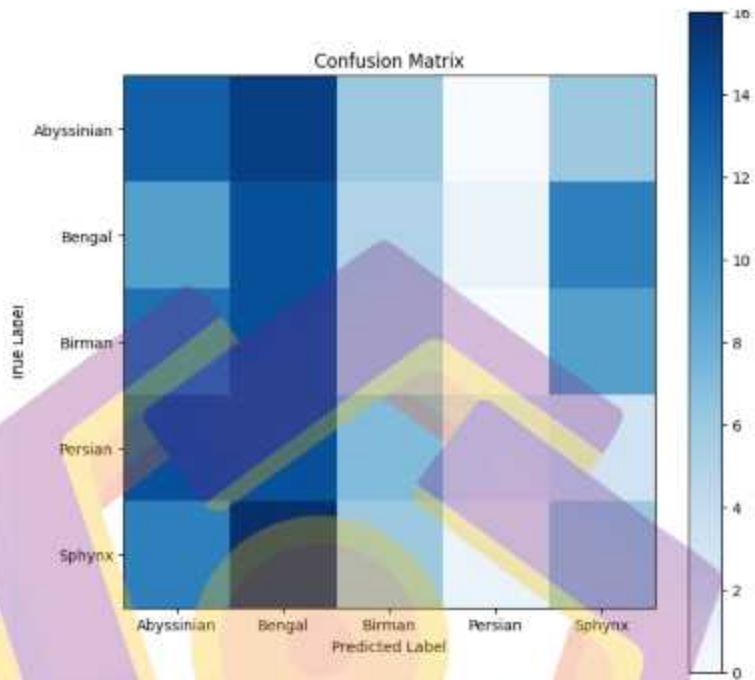
Dataset	Precision	Recall	F1 Score
Dataset Satu	0.14	0.19	0.15
Dataset Dua	0.25	0.2	0.18

Dalam penelitian ini, metrik evaluasi kinerja model seperti Precision, Recall, dan F1 Score digunakan untuk mengukur seberapa baik model CNN dapat mengidentifikasi citra kucing. Hasil penelitian menunjukkan bahwa model CNN mampu mencapai nilai-nilai yang dapat diinterpretasikan sebagai ukuran akurasi dan keandalan model dalam membedakan citra kucing dari citra lainnya. Model CNN mampu memberikan hasil yang dapat diandalkan dalam klasifikasi citra kucing, dengan metrik evaluasi yang memberikan gambaran komprehensif tentang kinerja model. Precision untuk Dataset Dua lebih tinggi dari Dataset Satu, namun F1 Score-nya lebih rendah. Ini menunjukkan bahwa meskipun model dapat mengidentifikasi lebih banyak kasus positif dengan benar dalam Dataset Dua, namun secara keseluruhan keseimbangan antara Precision dan Recall adalah lebih baik untuk Dataset Satu.



Gambar 4.11. Confusion Matriks Dataset Satu

Gambar 4.11 menunjukkan hasil dari evaluasi model confusion matriks, hasil tersebut menunjukkan kelas Sphynx lebih banyak mencapai tingkat akurasi hasil True Positif Lebih baik dari kelas lainnya.



Gambar 4.12. Confusion Matriks Dataset Dua

Gambar 4.12 menunjukkan model kedua hasil dari evaluasi model confusion matriks, hasil tersebut menunjukkan kelas Abbsyinian dan bengal lebih banyak mencapai tingkat akurasi hasil True Positif Lebih baik dari kelas lainnya.

## **BAB V**

### **PENUTUP**

#### **5.1. Kesimpulan**

Penelitian ini menggunakan metode Convolutional Neural Network (CNN) untuk mengklasifikasikan citra kucing dengan tingkat akurasi model CNN tertinggi ada pada pengujian dataset satu yang terdiri dari 500 data citra kucing dengan performa akurasi tertinggi pada training adalah 92,75% dan loss pada 0.2527, sedangkan untuk data validasi akurasi tertinggi adalah 84% dan loss pada 0.6870. Berdasarkan hasil diatas maka dapat diambil kesimpulan bahwa performance yang terbaik didapatkan pada skenario pertama yaitu dengan jumlah data 500 citra.

#### **5.2. Saran**

Saran bagi penelitian selanjutnya adalah dapat memperluas penelitian klasifikasi citra kucing menggunakan deep learning: convolutional neural network (CNN) serta dapat menggabungkannya dengan algoritma data mining lain seperti K-Means atau Mean Shift untuk mengidentifikasi kelas atau kelompok citra kucing yang serupa sebelum memasukkan data ke dalam model CNN. Ini dapat membantu dalam memperbaiki kualitas dan struktur dataset, serta mengurangi noise data.

## DAFTAR PUSTAKA

### PUSTAKA BUKU

Glauner, P.O., (2015). Comparison of Training Methods for Deep Neural Networks. ArXiv, abs/1504.06825.

Suyanto, (2018), Machine Learning Tingkat Dasar dan Lanjut, Penerbit Informatika Bandung.

Werdiningsih, L., Nuqoba, B. & Muhammadun. (2020). Data Mining Menggunakan Android, Weka, dan SPSS. Surabaya: Airlangga University Press.

Adinugroho, S., Sari, Y.A. (2018). Implementasi Data Mining Menggunakan Weka. Malang: UB Press.

### PUSTAKA MAJALAH, JURNAL ILMIAH ATAU PROSIDING

Mahmud, K. H., Adiwijaya, Al Faraby, S. (2019). Klasifikasi Citra Multi-Kelas Menggunakan Convolutional Neural Network. e-Proceeding of Engineering : Vol.6, No.1 April 2019(pp. 2127)

Alamsyah (2020) . Mengklasifikasikan ekspresi citra wajah dengan algoritma Convolutional Neural Network (CNN)

Ferdi Maylani, Sriyanto, Nosiell, (2021) . Implementasi metode Data Mining untuk Memprediksi Warna Anak Kucing Pada Proses Pengembangbiakan Kucing Ras Menggunakan Algoritma Support Vector Machine (SVM). Muhammad Afif Amanullah Fawwas, Kurniawan Nur Ramadhani, S.T., MT,

Febriyanti Sthevanie, S.T., M.T (2021). Klasifikasi ras pada kucing menggunakan Convolutional Neural Network (CNN)

Salim, S. D., & Suryadibrata, A. 2019. Klasifikasi Anjing dan Kucing menggunakan Algoritma Linear.

Putra (2020). Menerapkan algoritma CNN dalam mengklasifikasikan penyakit retinopati yang dapat menyebabkan kebutaan terhadap mata.

### PUSTAKA ELEKTRONIK

Sakinah Indriyani, Febriyanti Sthevanie, Kurniawan Nur Ramadhani, "Pengenalan Ras Kucing Scottish Fold Menggunakan Metode Histogram of Oriented Gradients dan Jaringan Saraf Tiruan" e-Proceeding of Engineering : Vol.6,






No.2 Agustus 2019, doi:


<https://openlibrarypublications.telkomuniversity.ac.id/index.php/engineering/article/view/9961/9817>

- A.Ramadhayani, V.Lusiana, "Klasifikasi Jenis Kucing Menggunakan Algoritma Principal Component Analysis Dan K-Nearest Neighbor" jurnal informasi dan Komputer Vol: 10 No:2.2022, doi: <https://doi.org/10.35959/jik.v10i2.333>
- A.Riyadi, I.Wardhani dan S.Widayati, "Klasifikasi Citra Anjing Dan Kucing Menggunakan Metode Convolutional Neural Network (CNN)" Seminar Nasional Teknologi Informasi dan Komunikasi STI&K (SeNTIK) Volume 5 Nomor 1, 28 Agustus 2021, ISSN : 2581-2327 , <https://ejournal.jak-stik.ac.id/index.php/sentik/article/view/3256>
- M.Effendi, "Sistem Deteksi Wajah Jenis Kucing Dengan Image Classification Menggunakan OpenCV", <http://journal.thamrin.ac.id/index.php/jtik/article/view/283>
- N.Qatrunnada, "Klasifikasi Ras Kucing Dengan Convolutional Neural Network Pada Citra Multi Objek", [https://repository.unsri.ac.id/61998/1/RAMA\\_55201\\_09021181722067\\_0222\\_058001\\_0022127804\\_01\\_front\\_ref.pdf](https://repository.unsri.ac.id/61998/1/RAMA_55201_09021181722067_0222_058001_0022127804_01_front_ref.pdf)
- T.Adriyanto, R.Ramadhani, R.Helilintar, A.Riskyawana, "Classification of dog and cat images using the CNN method" ILKOM Jurnal Ilmiah Vol. 14, No. 3, December 2022, pp. 203-208 Accredited 2nd by RISTEKBRIN No. 200/M/KPT/2020; E-ISSN 2548-7779 | P-ISSN 2087-1716, <https://jurnal.fikom.umi.ac.id/index.php/ILKOM/article/view/1116>
- T.Jajodia I, P.Garg "Image Classification – Cat and Dog Images", International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056 Volume: 06 Issue: 12 | Dec 2019, <https://www.irjet.net/archives/V6/i12/IRJET-V6i1271.pdf>
- J.Kusuma, A. Jinan, M. Lubis, Rubianto, R. Rosnelly "Komparasi Algoritma Support Vector Machine Dan Naive Bayes Pada Klasifikasi Ras Kucing", <https://core.ac.uk/download/pdf/492765248.pdf>
- Azahro Choirunisa, N., Karlita, T., & Asmara, R. Deteksi Ras Kucing Menggunakan Compound Model Scaling Convolutional Neural Network. Technomedia Journal, 6(2), 236-251. <https://ijc.ilearning.co/index.php/TMJ/article/view/1704>

- S. La Ode Ansyarullah S., "Klasifikasi Cats dan Dogs dengan Metode CNN dalam Fungsi Aktivasi relu, sigmoid, softmax, softplus, softsign, dan selu", <https://prosiding.konik.id/index.php/konik/article/view/72>
- A. Muhammad Afif, R. Kurniawan Nur.2, S. Febryanti., "Klasifikasi Ras pada Kucing menggunakan Algoritma Convolutional Neural Network(CNN)", ISSN : 2355-9365 e-Proceeding of Engineering : Vol.8, No.1 Februari 2021 | Page 715, <https://openlibrarypublications.telkomuniversity.ac.id/index.php/engineering/article/view/14320/14104>
- D. Diffran Nur Cahyo1, F. Muhammad Anwar, N. Jangkung Tri, Kusriani, "Analisis Perbandingan Optimizer pada Arsitektur NASNetMobile Convolutional Neural Network untuk Klasifikasi Ras Kucing", doi: <https://doi.org/10.34151/jurtek.v15i2.4025>
- S. Fadhilah Gusti, P. Agi, W. Merlinda, "Classification Of Cat Sounds Using Convolutional Neural Network (CNN) And Long Short-Term Memory (LSTM) Methods", Jurnal Teknik Informatika (JUTIF) Vol. 3, No. 5, Oktober 2022, hlm. 1349-1353, doi: <https://doi.org/10.20884/1.jutif.2022.3.5.373>
- S. Indriyani, F. Sthevanie, KN. Ramadhani, "Pengenalan Ras Kucing Scottish Fold Menggunakan Metode Histogram of Oriented Gradients dan Jaringan Saraf Tiruan", ISSN : 2355-9365
- CT. Emanuella, Musfita, A. Lawi, " klasifikasi suara kucing dan anjing menggunakan convolutional network", Konferensi Nasional Ilmu Komputer (KONIK) 2021
- GA Anwar, D. Riminahsih, " Klasifikasi citra genus panthera menggunakan metode convolutional neural network (CNN)", <http://dx.doi.org/10.35760/ik.2019.v24i3.2364>

### LAMPIRAN




NO	RAS	NAMA KUCING	GAMBAR
1	Abyssinian	Abyssinian_1	 A photograph of an Abyssinian cat lying down on a dark surface. The cat has a ruddy coat with distinct tabby markings and large, alert ears.
2	Abyssinian	Abyssinian_2	 A photograph of an Abyssinian cat sitting upright on a dark surface. The cat has a ruddy coat with tabby markings and large, alert ears.
3	Abyssinian	Abyssinian_3	 A photograph of an Abyssinian cat sitting upright on a dark surface. The cat has a ruddy coat with tabby markings and large, alert ears.

4	Abyssinian	Abyssinian_4	
5	Abyssinian	Abyssinian_5	
6	Abyssinian	Abyssinian_6	
7	Abyssinian	Abyssinian_7	
8	Abyssinian	Abyssinian_8	

9	Abyssinian	Abyssinian_9		
10	Abyssinian	Abyssinian_10		

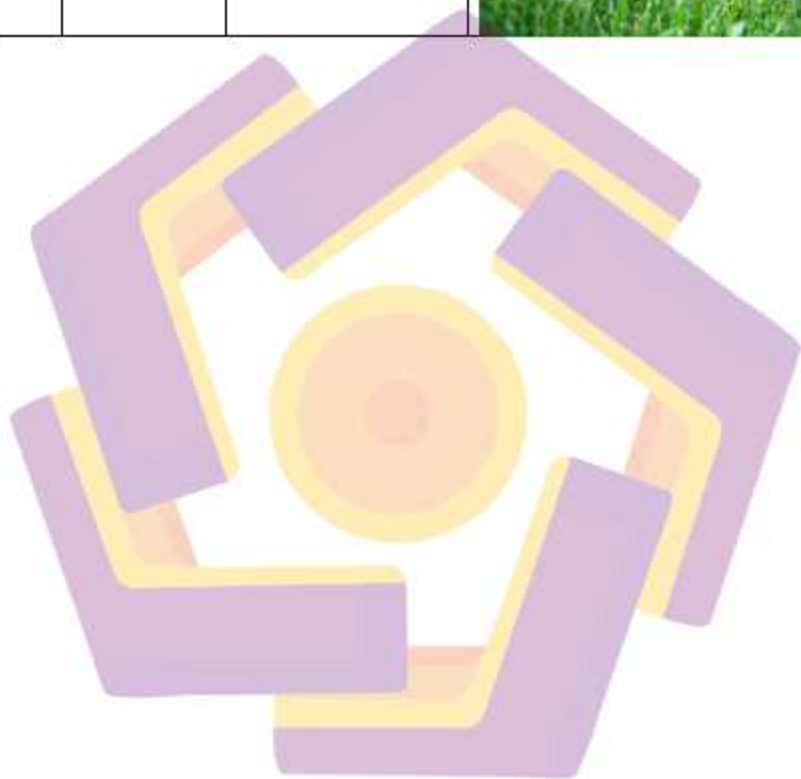






NO	RAS	NAMA KUCING	GAMBAR
1	Bengal	Bengal_1	
2	Bengal	Bengal_2	
3	Bengal	Bengal_3	

4	Bengal	Bengal_4	
5	Bengal	Bengal_5	
6	Bengal	Bengal_6	
7	Bengal	Bengal_7	
8	Bengal	Bengal_8	
9	Bengal	Bengal_9	






10	Bengal	Bengal_10	
----	--------	-----------	--






NO	RAS	NAMA KUCING	GAMBAR
1	Birman	Birman_1	
2	Birman	Birman_2	
3	Birman	Birman_3	
4	Birman	Birman_4	
5	Birman	Birman_5	




6	Birman	Birman_6	
7	Birman	Birman_7	
8	Birman	Birman_8	
9	Birman	Birman_9	
10	Birman	Birman_10	



NO	RAS	NAMA KUCING	GAMBAR
1	Persian	Persian_1	
2	Persian	Persian_2	
3	Persian	Persian_3	
4	Persian	Persian_4	
5	Persian	Persian_5	

6	Persian	Persian_6	
7	Persian	Persian_7	
8	Persian	Persian_8	
9	Persian	Persian_9	
10	Persian	Persian_10	

NO	RAS	NAMA KUCING	GAMBAR
1	Sphynx	Sphynx_1	
	Sphynx	Sphynx_2	
3	Sphynx	Sphynx_3	
4	Sphynx	Sphynx_4	
5	Sphynx	Sphynx_5	

6	Sphynx	Sphynx_8	
7	Sphynx	Sphynx_9	
8	Sphynx	Sphynx_11	
9	Sphynx	Sphynx_12	
10	Sphynx	Sphynx_14	