

**SISTEM DETEKSI EKSPRESI WAJAH BERBASIS
CONVOLUTIONAL NEURAL NETWORK (CNN) UNTUK
PENGENALAN EMOSI MANUSIA**

SKRIPSI

Diajukan sebagai Salah Satu Syarat untuk Mendapatkan
Gelar Sarjana Komputer (S.kom) Program Studi Informatika



**ARYA WIBAWA. AR
105841109320**

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

**SISTEM DETEKSI EKSPRESI WAJAH BERBASIS
CONVOLUTIONAL NEURAL NETWORK (CNN) UNTUK
PENGENALAN EMOSI MANUSIA**

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

Disusun dan Diajukan Oleh :

ARYA WIBAWA. AR

1058411109320

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



UNIVERSITAS MUHAMMADIYAH MAKASSAR

FAKULTAS TEKNIK

GEDUNG MENARA IQRA LT. 3

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

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

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

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

PENGESAHAN

Skripsi atas nama Arya Wibawa Ar. dengan nomor Induk Mahasiswa 105 84 11093 20, dinyatakan diterima dan disahkan oleh Panitia Ujian Tugas Akhir/Skripsi sesuai dengan Surat Keputusan Dekan Fakultas Teknik Universitas Muhammadiyah Makassar Nomor : 105/05/A.5-VIN/45/2024, sebagai salah satu syarat guna memperoleh gelar Sarjana Komputer pada Program Studi Informatika Fakultas Teknik Universitas Muhammadiyah Makassar pada hari Sabtu tanggal 24 Agustus 2024.

Panitia Ujian :

1. Pengawas Umum

Makassar, - 19 Safar 1446 H
- 24 Agustus 2024 M

a. Rektor Universitas Muhammadiyah Makassar

Dr. Ir. H. Abd. Rakhim Nanda, ST., MT., IPU. :

b. Dekan Fakultas Teknik Universitas Hasanuddin

Prof. Dr. Eng. Muhammad Isran Ramli, ST., MT. :

2. Penguji

a. Ketua : Dr. Ir. Hj. Hafsa Niwaria, ST., MT. :

b. Sekretaris : Lukman, S.Kom., M.T. :

3. Anggota : 1. Muhyiddin A.M. Hayat, S.Kom., M.T. :

2. Desi Anggreani, S.Kom., MT. :

3. Titin Wahyuni, S.Pd., M.T. :

Mengetahui :

Pembimbing I

Pembimbing II

Fahrim Irhamna Rahman S.Kom., M.T.

Rizki Yusliana Bakti ST., MT

Dekan





UNIVERSITAS MUHAMMADIYAH MAKASSAR

FAKULTAS TEKNIK

GEDUNG MENARA IQRA LT. 3

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

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

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

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

HALAMAN PENGESAHAN

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

Judul Skripsi : SISTEM DETEKSI EKPRESI WAJAH BERBASIS
CONVOLUTIONAL NEURAL NETWORK (CNN) UNTUK
PENGENALAN EMOSI MANUSIA

Nama : Arya Wibawa. Ar

Stambuk : 105 84 11093 20

Makassar, 30 Agustus 2024

Telah Diperiksa dan Disetujui
Oleh Dosen Pembimbing;

Pembimbing I

Fahrim Irhamna Rahman S.Kom., MT.

Pembimbing II

Rizki Yusliana Bakti, ST., MT.

Mengetahui,

Ketua Program Studi Informatika



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

NBM. 1504 577

ABSTRAK

ARYA WIBAWA. AR. Sistem Deteksi Ekspresi Wajah Berbasis Convolutional Neural Network (CNN) Untuk Pengenalan Emosi Manusia (Dibimbing Oleh Fahrin Irhamna Rachman S.Kom., M.T., dan Rizki Yusliana Bakti S.T.,M.T.,).

Pengenalan sistem deteksi ekspresi wajah manusia telah menjadi topik penelitian yang semakin berkembang, terutama dalam upaya mengembangkan aplikasi yang mampu memahami dan merespons emosi manusia secara otomatis. Pada penelitian ini bertujuan untuk mengembangkan dan mengevaluasi sistem deteksi ekspresi wajah manusia menggunakan metode *Convolutional Neural Network (CNN)*. Dataset yang digunakan berjumlah 800 dataset terdiri dari gambar wajah dengan ekspresi senang, marah, sedih dan takut. Data ini diproses melalui beberapa tahap preprocessing, termasuk normalisasi, augmentasi, dan pembagian data menjadi set latih dan uji. Penelitian ini menggunakan beberapa arsitektur *CNN* untuk mengidentifikasi emosi seperti senang, sedih, marah, dan takut. Pengujian dilakukan menggunakan berbagai parameter, termasuk pembagian data latih dan uji, serta arsitektur *CNN* yang berbeda. Hasil penelitian menunjukkan bahwa model *CNN* mampu mencapai akurasi lebih dari 90% pada data pelatihan, dengan performa terbaik pada emosi "Senang" dengan *f1-score* 0.93. Namun, terdapat penurunan akurasi pada data validasi, dengan rata-rata akurasi keseluruhan sebesar 78%, menunjukkan adanya tantangan dalam generalisasi model. Selain itu, emosi "Sedih" memiliki *recall* terendah sebesar 0.49, mengindikasikan perlunya peningkatan model dalam klasifikasi emosi tertentu. Penelitian ini memberikan kontribusi dalam pengembangan sistem deteksi ekspresi wajah berbasis *CNN*, namun diperlukan eksplorasi lebih lanjut terhadap arsitektur yang lebih kompleks, evaluasi dengan dataset yang lebih beragam, dan pengujian pada data real-time untuk meningkatkan performa sistem.

Kata Kunci: Ekspresi, Deteksi, dan *Convolutional Neural Network (CNN)*

ABSTRACT

ARYA WIBAWA. AR. *Facial Expression Detection System Based on Convolutional Neural Network (CNN) for Human Emotion Recognition (Supervised by Fahrira Irhamna Rachman S.Kom., M.T., dan Rizki Yusliana Bakti S.T.,M.T.,).*

The introduction of human facial expression detection systems has become an increasingly popular research topic, especially in the effort to develop applications capable of automatically understanding and responding to human emotions. This study aims to develop and evaluate a human facial expression detection system using the Convolutional Neural Network (CNN) method. The dataset used consists of 800 images of faces expressing happiness, anger, sadness, and fear.. This data undergoes several preprocessing stages, including normalization, augmentation, and splitting into training and testing sets. The study employs multiple CNN architectures to identify emotions such as happiness, sadness, anger, and fear. Testing is conducted using various parameters, including different training and testing data splits, as well as different CNN architectures. The results show that the CNN model achieved over 90% accuracy on the training data, with the best performance on the "Happiness" emotion, yielding an F1-score of 0.93. However, there was a decrease in accuracy on the validation data, with an overall average accuracy of 78%, indicating challenges in model generalization. Additionally, the "Sadness" emotion had the lowest recall at 0.49, suggesting the need for model improvement in classifying certain emotions. This research contributes to the development of CNN-based facial expression detection systems; however, further exploration of more complex architectures, evaluation with more diverse datasets, and testing on real-time data are necessary to improve system performance.

Keywords: *Expression, Detection, Convolutional Neural Network (CNN)*

KATA PENGANTAR

الرَّحِيمِ الرَّحْمَنِ اللَّهُ بِسْمِ

Assalamu'alaikum Warahmatullahi Wabaraktuh

Segala puji bagi Allah Subhanallahu Wa Ta'ala atas limpahan rahmat dan karunianya sehingga penulis dapat menyelesaikan Laporan Tugas Akhir yang berjudul “SISTEM DETEKSI EKSPRESI WAJAH BERBASIS *CONVOLUTIONAL NEURAL NETWORK (CNN)* UNTUK PENGENALAN EMOSI MANUSIA”. Salawat beserta Salam senantiasa penulis panjatkan kepada Nabi Muhammad SAW, yang telah membawa kita dari Zaman jahiliah menuju Zaman yang serba modern seperti saat ini.

Tak lupa penulis mengucapkan banyak-banyak terima kasih kepada seluruh pihak yang telah membantu dan memberikan wejangan dalam penyusunan skripsi ini, terutama kepada:

1. Terkhusus kedua orang tuaku tercinta **Bapak Ali Rahman** dan **Ibu Rostini**, terima kasih sudah mengajarku banyak hal, memberikanku inspirasi, doa, dan dukungan serta jasa- jasa yang tidak akan mampu saya ganti hingga akhir hayat. Semoga saya bisa berikan yang terbaik.
2. Ibu **Dr.Ir.Hj Nurnawati, S.T., M.T., I.P.M**, selaku Dekan Fakultas Teknik.
3. Bapak **Muh. Syafaat S Kuba, S.T., M.T**, selaku Wakil Dekan 1 Fakultas Teknik.
4. Bapak **Muhyiddin AM Hayat S.Kom., M.T**, selaku Ketua Prodi Informatika.
5. Bapak **Fahrim Irhamna Rachman S.Kom., M.T**, selaku Dosen Pembimbing 1.
6. Ibu **Rizki Yusliana Bakti S.T.,M.T** selaku Dosen Pembimbing 2.
7. Untuk kakakku Arni Evayani dan Andri Puasmanto, terima kasih sudah membiayai, mendukung, dan mendoakan untuk saya hingga saat ini.
8. Dosen dan Staf Fakultas Teknik Universitas Muhammadiyah Makassar.

9. Sahabatku David Arian Virgiawan, Akram, Muhammad Fachri Rasyidi, Syamsul Alam, Muh. Firdaus, Annisa Tiara Saputri Syatir, Rosalinda Aprilia Sari, Lis Indriani, Ayu Andira, dan Rizka Adrianingsih, terima kasih telah menemani, mendukung dan mendoakan untuk saya.
10. Sahabat dekatku Alexander Skevin Frandito Amantokan Nasus, Laode Muh. Taufiq Qurrahman, Esy Anugrah Rahayu Kasim, Ummi Kalsum, Pasha Diza Putri, Nurul Hijrah, dan Nadiyah Nurfadillah, terima kasih telah mendukung dan mendoakan untuk saya.
11. Teman-teman Khususnya Informatika kelas C 2020 Fakultas Teknik, Universitas Muhammadiyah Makassar, terima kasih atas dukungan dan doanya.

Demikian laporan skripsi ini, dan penulis sadar bahwa laporan ini masih banyak kekurangan di dalamnya oleh karena itu penulis mengharapkan kritik dan saran yang sifatnya membangun dari pembaca atas laporan ini. Akhir kata penulis ucapkan terima kasih.

Billahi fisabilhaq, fastabiqul khairat.

Wassalamualaikum Wr.Wb.

Makassar, Agustus 2024

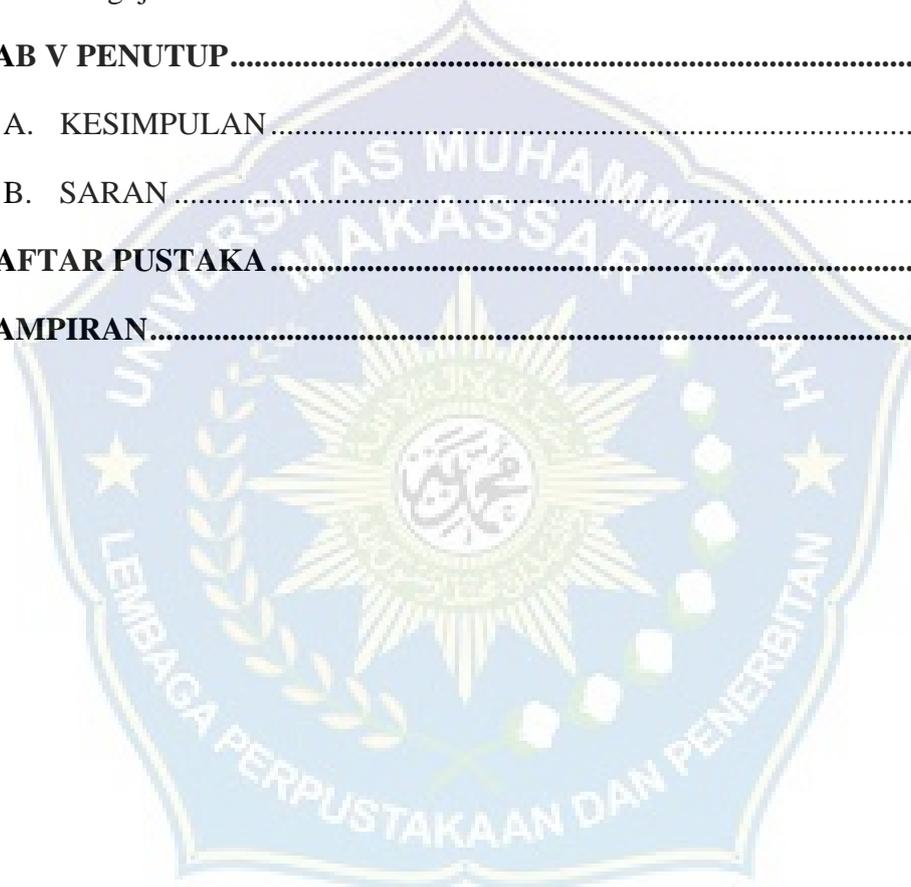
Penulis

Arya Wibawa. Ar

DAFTAR ISI

ABSTRAK	v
ABSTRACT	vi
KATA PENGANTAR.....	v
DAFTAR ISI.....	vii
DAFTAR GAMBAR.....	ix
DAFTAR TABEL.....	x
DAFTAR LAMPIRAN	xi
DAFTAR ISTILAH	xii
BAB I PENDAHULUAN.....	1
A. Latar Belakang	1
B. Rumusan Masalah	2
C. Tujuan Penelitian	2
D. Manfaat Penelitian	3
E. Ruang Lingkup Penelitian.....	3
F. Sistematika Penulisan.....	4
BAB II TINJAUAN PUSTAKA.....	5
A. Landasan Teori.....	5
B. Penelitian Terkait	12
C. Kerangka Pikir	18
BAB III METODE PENELITIAN	19
A. Tempat dan Waktu Penelitian	19
B. Bahan dan Alat Penelitian.....	19
C. Perancangan Sistem	19
D. Cara kerja CNN	22

E. Pengujian sistem.....	24
F. Teknik dan Analisis Data.....	26
BAB VI HASIL DAN PEMBAHASAN.....	27
A. Pembuatan Model.....	27
B. Training Googlecolab.....	31
C. Pengujian sistem.....	32
BAB V PENUTUP.....	38
A. KESIMPULAN.....	38
B. SARAN.....	39
DAFTAR PUSTAKA.....	40
LAMPIRAN.....	43

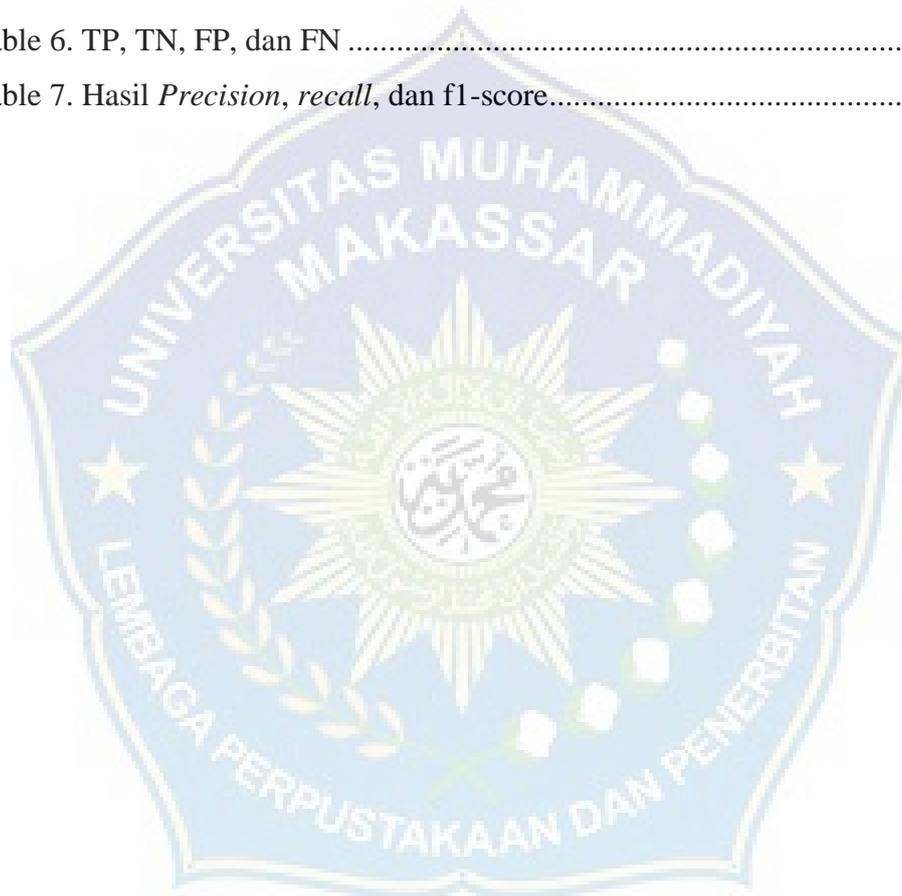


DAFTAR GAMBAR

Gambar 1. Gambar Ekspresi	5
Gambar 2. Ilustrasi Arsitektur Convosional Neural Network	7
Gambar 3. Ilustrasi Arsitektur <i>Convolutional Neural Network</i>	7
Gambar 4. Contoh Operasi Matematis Proses Konvolusi.....	8
Gambar 5. Contoh Penerapan Aktivasi ReLu	9
Gambar 6. Kerangka Pikir.....	18
Gambar 7. Cara Kerja CNN.....	22
Gambar 8. Contoh Ekspresi Senang.....	27
Gambar 9. Contoh Ekspresi Sedih	28
Gambar 10. Contoh Ekspresi Marah.....	28
Gambar 11. Contoh Ekspresi Takut.....	28
Gambar 12. Upload Dataset.....	29
Gambar 13. Class Dataset	29
Gambar 14. Pelabelan Dataset	30
Gambar 15. Preprocessing Gambar.....	30
Gambar 16. Proses Training.....	31
Gambar 17. Hasil Dari Training.....	31
Gambar 18. Gambar dengan prediksi bounding box	32
Gambar 19. Hasil Deteksi Gambar	33
Gambar 20. Prediksi dan Label Asli	35

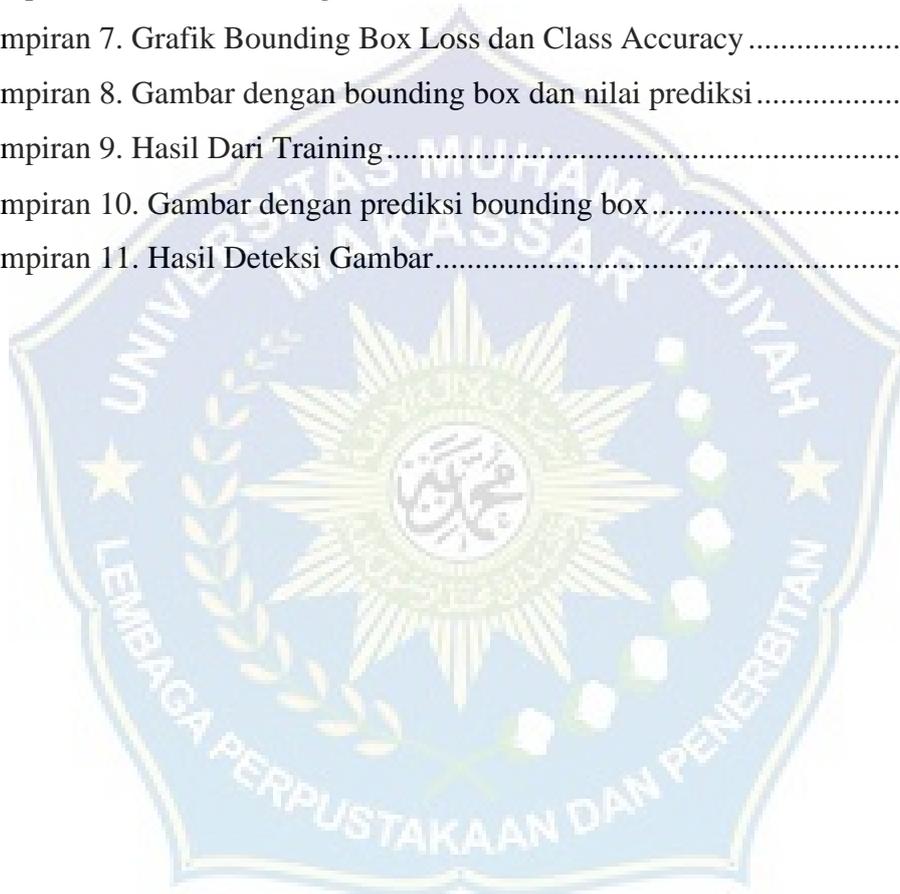
DAFTAR TABEL

Table 1. Fungsi <i>TensorFlow</i>	11
Table 2. Persamaan dan Perbedaan penelitian	15
Table 3. Jumlah Dataset	27
Table 4. Hasil Pengujian	34
Table 5. Identifikasi Prediksi dan Label Asli	35
Table 6. TP, TN, FP, dan FN	36
Table 7. Hasil <i>Precision</i> , <i>recall</i> , dan <i>f1-score</i>	37



DAFTAR LAMPIRAN

Lampiran 1. Dataset Marah.....	43
Lampiran 2. Dataset Sedih.....	44
Lampiran 3. Dataset Senang.....	46
Lampiran 4. Dataset Takut.....	47
Lampiran 5. Source Code.....	48
Lampiran 6. Proses Training.....	55
Lampiran 7. Grafik Bounding Box Loss dan Class Accuracy.....	55
Lampiran 8. Gambar dengan bounding box dan nilai prediksi.....	56
Lampiran 9. Hasil Dari Training.....	56
Lampiran 10. Gambar dengan prediksi bounding box.....	56
Lampiran 11. Hasil Deteksi Gambar.....	57



DAFTAR ISTILAH

<i>Python</i>	<i>Python</i> merupakan bahasa pemrograman komputer yang biasa dipakai untuk membangun situs, software/aplikasi, mengotomatiskan tugas dan melakukan analisis data.
<i>Machine Learning</i>	<i>Machine Learning</i> dapat didefinisikan sebagai penerapan teknologi komputer dan algoritma matematika yang menggunakan pendekatan pembelajaran dari data untuk menghasilkan prediksi di masa depan
<i>Deep learning</i>	<i>Deep learning</i> adalah metode dalam kecerdasan buatan (AI) yang mengajarkan komputer untuk memproses data dengan cara yang terinspirasi otak manusia.
<i>Convolutional Neural Network (CNN)</i>	<i>Convolutional Neural Network (CNN)</i> merupakan salah satu kelas dari <i>Deep learning</i> yang mampu melakukan pengenalan gambar dan klasifikasi gambar.
<i>Keras</i>	<i>Keras</i> adalah library open source untuk jaringan syaraf yang ditulis dalam bahasa pemrograman <i>Python</i> .
<i>Object Detection</i>	Mereplikasi kecerdasan yang dimiliki manusia dalam melihat benda menggunakan komputer
<i>TensorFlow</i>	<i>TensorFlow</i> adalah library perangkat lunak open-source yang sering digunakan untuk dataflow dan pemrograman diferensial dalam berbagai tugas.

TensorFlow

TensorFlow atau sering disebut dengan diagram alir merupakan suatu jenis diagram yang merepresentasikan algoritma atau langkah-langkah instruksi yang berurutan dalam sistem.

Roboflow

Roboflow adalah kerangka kerja pengembang *computer vision* untuk pengumpulan data yang lebih baik hingga prapemrosesan, dan teknik pelatihan model.



BAB I

PENDAHULUAN

A. Latar Belakang

Wajah adalah fitur utama yang digunakan untuk mengenali seseorang. Ekspresi wajah manusia mencerminkan emosi dan perasaan mereka saat berinteraksi sosial. Ekspresi wajah dapat mengungkapkan emosi yang dirasakan seseorang (Widodo et al., 2022). Ekspresi wajah adalah bentuk komunikasi non-verbal yang sangat penting dalam interaksi manusia. Wajah sering kali menjadi refleksi dari emosi, perasaan, dan pikiran seseorang. Melalui ekspresi wajah, seseorang bisa menunjukkan kebahagiaan, kesedihan, ketakutan, atau berbagai emosi lainnya tanpa harus mengucapkan kata-kata. Dengan memahami emosi yang dialami seseorang, kita dapat menilai kondisi psikologisnya (Achmad et al., 2019)). Inilah yang menjadikan ekspresi wajah sangat penting dalam berbagai bidang, termasuk dalam pengembangan teknologi robotika.

Dengan kemajuan teknologi pengolahan citra dan kecerdasan buatan, menjadi mungkin untuk mengembangkan sistem yang lebih akurat dan efisien dalam mendeteksi ekspresi wajah. Salah satu metode yang telah terbukti efektif adalah penggunaan *Convolutional Neural Network (CNN)*, yang memiliki tingkat keakuratan tinggi (Muttaqiin et al., 2023). *CNN* ialah jenis arsitektur jaringan saraf tiruan yang sangat efektif untuk mengenali pola dalam data gambar, termasuk ekspresi wajah (Fasounaki et al., 2021).

CNN dirancang khusus untuk mengenali dan mengklasifikasikan gambar. *CNN* terdiri dari beberapa lapisan (layer) yang berfungsi mengekstrak informasi dari gambar dan menentukan klasifikasi gambar dalam bentuk skor klasifikasi (Nugroho et al., 2020). Sistem deteksi ekspresi wajah berbasis *CNN* memiliki berbagai aplikasi penting. Dalam interaksi manusia-komputer, teknologi ini memungkinkan pengembangan antarmuka yang lebih intuitif dan responsif, yang dapat menyesuaikan perilaku sistem berdasarkan emosi pengguna.

Ke depannya, dengan pengembangan dan penerapan teknologi *CNN* dalam interpretasi ekspresi wajah, diharapkan dapat memberikan kontribusi positif dalam berbagai bidang.

Di bidang kesehatan mental, teknologi ini dapat membantu dalam pemantauan dan diagnosis kondisi psikologis dengan mengidentifikasi tanda-tanda emosi tertentu yang terkait dengan gangguan mental. Dalam pendidikan, sistem ini dapat digunakan untuk mengevaluasi respons emosional siswa selama proses pembelajaran, membantu guru dalam menyesuaikan metode pengajaran. Selain itu, dalam sistem pengawasan, teknologi ini dapat meningkatkan keamanan dengan mendeteksi tanda-tanda perilaku mencurigakan atau stres pada orang-orang di area publik. Penelitian dan pengembangan lebih lanjut dalam bidang ini diharapkan dapat membuka peluang baru untuk aplikasi yang lebih luas dan inovatif, membawa kita lebih dekat kepada pemahaman yang lebih dalam tentang emosi manusia melalui ekspresi wajah.

B. Rumusan Masalah

Berlandaskan latar belakang diatas, peneliti dapat merumuskan masalah sebagai berikut.

1. Bagaimana penerapan metode *CNN* pada sistem deteksi ekspresi wajah?
2. Bagaimana akurasi dari sistem yang mendeteksi ekspresi manusia dengan menggunakan metode *CNN*?

C. Tujuan Penelitian

Berdasarkan Rumusan Masalah diatas dapat disimpulkan bahwa tujuan penelitian ini adalah sebagai berikut:

1. Untuk mengetahui penerapan metode *CNN* pada sistem deteksi ekspresi wajah.
2. Untuk mengetahui akurasi dari sistem yang mendeteksi ekspresi manusia dengan menggunakan metode *CNN*.

D. Manfaat Penelitian

Hasil penelitian ini diharapkan dapat memberikan manfaat sebagai berikut:

1. Bagi bidang robotik
 - a. Robot yang mampu mengenali dan memahami ekspresi wajah dapat berinteraksi dengan manusia secara lebih alami dan responsif.
 - b. Dengan kemampuan untuk mengidentifikasi emosi manusia, robot dapat memberikan respons yang lebih tepat dan sesuai dengan kondisi emosional pengguna.
 - c. Robot yang dilengkapi dengan teknologi pengenalan ekspresi wajah dapat digunakan dalam berbagai layanan seperti perawatan kesehatan, pendidikan, dan layanan pelanggan.
2. Bagi peneliti
 - a. Penelitian ini membuka peluang untuk mengembangkan teknologi baru dalam pengenalan ekspresi wajah, memberikan dorongan untuk inovasi di bidang kesehatan mental.
 - b. Penelitian ini dapat memberikan kontribusi pada ilmu pengetahuan dengan memperluas pemahaman tentang hubungan antara teknologi dan konseling psikologi.
 - c. Hasil penelitian membantu peneliti memahami cara terbaik memanfaatkan teknologi *CNN* untuk mendukung konseling psikologi.

E. Ruang Lingkup Penelitian

Adapun ruang lingkup penelitian pada penelitian ini adalah sebagai berikut:

1. Proses pengumpulan dan penyusunan dataset yang terdiri dari gambar ekspresi wajah senang, sedih, marah, dan takut.
2. Teknik pra-pemrosesan citra yang digunakan untuk mempersiapkan gambar berukuran 320 x 320 piksel.
3. Penyesuaian metode deteksi ekspresi wajah dengan karakteristik pengambilan gambar melalui kamera.
4. Penekanan pada penggunaan gambar dalam format 2D untuk deteksi ekspresi wajah.

5. Proses pelatihan model *CNN* menggunakan dataset yang disiapkan.
6. Menggunakan model *MobileNet*.

F. Sistematika Penulisan

Berikut ini adalah gambaran umum dari keseluruhan penulisan:

BAB I PENDAHULUAN

Bab ini mencakup latar belakang penelitian, perumusan masalah, tujuan yang ingin dicapai, manfaat penelitian, ruang lingkup, serta penjelasan mengenai struktur penulisan yang akan dibahas.

BAB II TINJAUAN PUSTAKA

Bab ini membahas literatur yang relevan, termasuk teori-teori dasar dan kerangka konseptual yang mendasari penelitian.

BAB III METODE PENELITIAN

Bab ini menguraikan secara rinci metode penelitian yang akan digunakan, mencakup langkah-langkah dalam mengidentifikasi masalah, proses pengumpulan data, analisis algoritma, serta penggunaan instrumen yang tepat dalam penelitian.

BAB II

TINJAUAN PUSTAKA

A. Landasan Teori

1. Ekspresi Wajah

Wajah ialah salah satu bagian yang paling khas dari tubuh manusia. Ini adalah area yang paling biasa di amati dan di komunikasikan dalam interaksi sosial. Ekspresi wajah adalah isyarat yang sering digunakan manusia untuk mengkomunikasikan perasaan, pikiran, dan niat mereka kepada orang lain.

Menurut (KBBI) Online, n.d.) Wajah adalah bagian depan kepala manusia yang mencakup area dari dahi hingga dagu, termasuk rambut, dahi, alis, pelipis, mata, hidung, pipi, mulut, bibir, gigi, kulit, dan dagu. Wajah terutama berfungsi untuk ekspresi wajah, penampilan, serta sebagai identitas seseorang, atau dalam hal ini Pengenalan ekspresi wajah adalah proses teknologi atau metode di mana sistem komputer dilatih untuk mengidentifikasi dan memahami ekspresi wajah manusia (Syahromi, 2022). Teknologi ini menggunakan algoritma pengolahan citra dan visi komputer untuk menganalisis ciri wajah seseorang, seperti bentuk mata, posisi alis, gerak bibir, dan lainnya, untuk kemudian mengklasifikasikan ekspresi wajah ke dalam kategori emosi tertentu seperti kebahagiaan, kesedihan, kemarahan, atau ketakutan.



Gambar 1. Gambar Ekspresi (Segal, 2022)

2. *Deep learning*

Deep learning adalah subbidang dari *Machine Learning* yang menggunakan *neural networks* dengan banyak layer (lebih dari 3) untuk mempelajari representasi data yang semakin kompleks dan abstrak. *Deep learning* telah menunjukkan keunggulan dalam berbagai tugas seperti pengenalan citra, pengenalan suara, dan pemrosesan bahasa alami. Salah satu teknik *Deep learning* yang populer adalah *CNN* yang sering digunakan dalam pengolahan citra (Peryanto et al., 2020).

3. Deteksi Objek

Deteksi objek (*Object Detection*) adalah teknik visi komputer untuk menemukan contoh objek dalam gambar atau video. Algoritma deteksi objek biasanya memanfaatkan pembelajaran mesin atau pembelajaran mendalam untuk menghasilkan hasil yang bermakna. Ketika manusia melihat gambar atau video, manusia dapat mengenali dan menemukan objek dalam beberapa saat berbeda dengan komputer yang memerlukan komputasi yang kompleks. Tujuan deteksi objek adalah untuk mereplikasi kecerdasan yang dimiliki manusia dalam melihat benda menggunakan komputer. Cara kerja deteksi objek adalah deteksi objek menempatkan keberadaan objek dalam gambar (Komputasi et al., 2020).

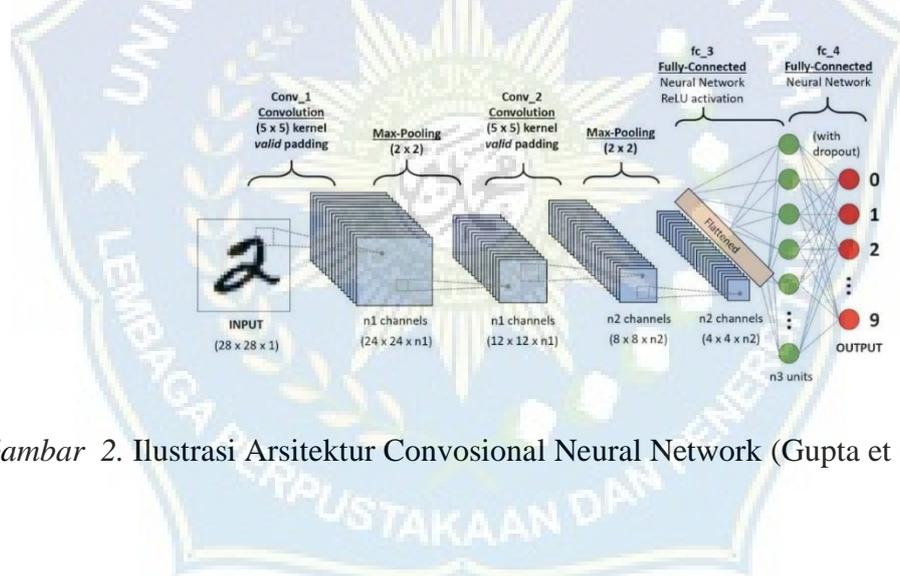
Dalam proses deteksi objek, sistem menerima gambar sebagai input dan memberikan respons dengan menampilkan hasil klasifikasi yang sesuai dengan gambar tersebut. Secara umum, deteksi objek digunakan untuk mengidentifikasi berbagai objek, seperti manusia, bangunan, atau mobil. Dengan kemajuan teknologi, deteksi objek memiliki beragam aplikasi dalam bidang visi komputer, yang termasuk dalam upaya meningkatkan otomatisasi (Tiyar & Fudholi, 2021).

4. *Convolutional Neural Network*

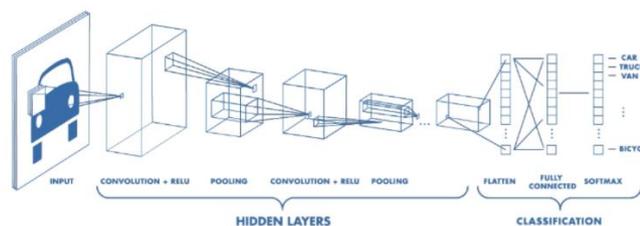
Convolutional Neural Network (CNN) adalah salah satu jenis *Deep learning* yang mampu mengenali dan mengklasifikasikan gambar. *CNN* merupakan jenis neural network yang khusus menangani data dengan struktur topologi seperti grid, seperti gambar. *CNN* dapat diterapkan dalam berbagai

tugas, seperti pengenalan wajah, analisis dokumen, klasifikasi gambar, klasifikasi video, dan lainnya. (Gupta et al., 2021).

Metode *CNN* mengklasifikasikan gambar dengan memproses gambar yang diinput, lalu mengelompokkannya ke dalam kategori tertentu, misalnya pada citra manusia, terdapat wajah, mata, bibir, hidung, tangan, dan lain-lain. Gambar diubah menjadi array yang berisi nilai-nilai setiap piksel dengan resolusi tinggi*lebar*dimensi, yang disebut sebagai channel. Biasanya, channel ini terdiri dari 3 lapisan yang menunjukkan citra sebagai gambar RGB, di mana setiap lapisan (channel) merepresentasikan warna Merah, Hijau, dan Biru (Red-Green-Blue), atau 1 lapisan jika gambar grayscale. Namun, jumlah lapisan ini bisa lebih dari 3, bahkan mencapai ratusan, yang merepresentasikan berbagai warna lainnya dalam arsitektur RGB.



Gambar 2. Ilustrasi Arsitektur Convosional Neural Network (Gupta et al., 2021)



Gambar 3. Ilustrasi Arsitektur Convolutional Neural Network (Sena, 2019)

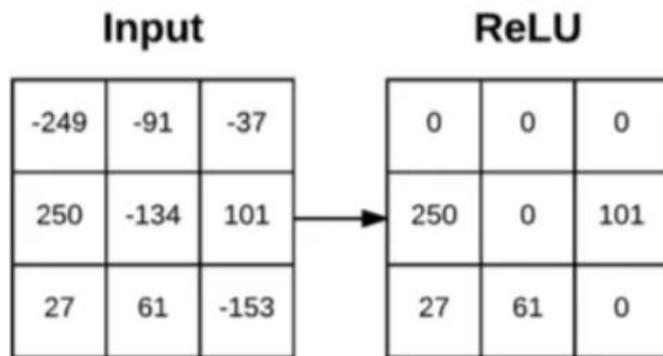
Seperti yang ditunjukkan pada Gambar 2 dan Gambar 3, arsitektur *CNN* terdiri dari dua bagian utama, yaitu Feature Extraction Layer dan Fully Connected Layer. Pada Feature Extraction Layer, gambar di-"encode" menjadi feature map, yaitu sekumpulan angka yang merepresentasikan citra tersebut (Feature Extraction).

Feature Extraction Layer terdiri dari convolution layer dan pooling layer. Convolution layer merupakan komponen utama dalam metode *CNN* yang membedakannya dari neural network lainnya. Convolution layer adalah lapisan pertama yang mengekstraksi fitur dari citra yang dimasukkan. Proses konvolusi (Convolution) mempertahankan hubungan antar piksel dengan mempelajari fitur citra melalui operasi matematis antara matriks citra dan filter atau kernel. (Gupta et al., 2021). Kernel adalah operator yang diterapkan pada seluruh citra untuk menghasilkan nilai array dari citra tersebut. Kernel adalah matriks, biasanya berukuran 3x3 atau 5x5, yang berisi nilai acak antara -1 dan 1. Contoh operasi matematis dalam proses konvolusi ditunjukkan pada Gambar 4.

131	162	232	84	91	207
104	-1	0	+1	237	109
243	-2	0	+2	185	26
185	-1	0	+1	61	225
157	124	25	14	102	108
5	155	116	218	232	249

Gambar 4. Contoh Operasi Matematis Proses Konvolusi (Syamsudin, 2019)

Hasil dari konvolusi antara matriks citra dan filter (kernel) disebut feature map. Selain itu, terdapat ReLU atau Rectified Linear Unit yang berfungsi untuk mengubah nilai negatif pada feature map menjadi positif, seperti yang ditunjukkan pada Gambar 5.



Gambar 5. Contoh Penerapan Aktivasi ReLu (Syamsudin, 2019)

Pooling layer, atau proses Max-Pooling, adalah lapisan yang berfungsi untuk mengurangi ukuran spasial guna mengurangi jumlah parameter dan perhitungan ketika ukuran citra terlalu besar. Pooling layer bekerja secara independen pada setiap feature map.

Fully Connected Layer adalah bagian di mana hasil feature map dari feature extraction layer, yang berupa array multi-dimensi, diubah menjadi vektor melalui proses yang disebut Flatten. Vektor ini kemudian digunakan untuk membentuk fully connected layer, mirip dengan jaringan syaraf (neural network), dan semua fitur yang telah dibentuk digabungkan untuk membangun model. (Gupta et al., 2021).

Secara ringkas, cara kerja metode *CNN* melibatkan pelatihan dan pengujian setiap gambar melalui serangkaian proses. Proses ini dimulai dengan memecah gambar menjadi bagian-bagian yang lebih kecil, kemudian memasukkan setiap bagian kecil ke dalam neural network yang lebih kecil. Hasil dari masing-masing bagian kecil disimpan dalam array baru, diikuti dengan downsampling atau pengurangan ukuran spasial untuk mengurangi jumlah parameter dan perhitungan saat ukuran citra terlalu besar, dan akhirnya membuat prediksi.

Kelebihan metode *CNN* meliputi kemampuannya untuk secara otomatis mengekstraksi fitur penting dari setiap citra tanpa intervensi manusia, serta efisiensinya dibandingkan dengan metode neural network lainnya, terutama dalam hal memori dan kompleksitas. Namun, kekurangan metode *CNN* meliputi kebutuhan akan data pelatihan yang besar, proses pelatihan yang

memakan waktu, dan risiko overfitting. Overfitting dapat terjadi karena dengan terlalu banyak data pelatihan, algoritma kehilangan kemampuan untuk menggeneralisasi.

5. Keras

Keras adalah pustaka open source untuk jaringan syaraf yang ditulis dalam bahasa pemrograman *Python*. *Keras* dapat dijalankan di Tensor Processing Unit (TPU) atau GPU yang besar, dan model yang dibangun dengan *Keras* dapat diekspor untuk digunakan di browser.

6. TensorFlow

TensorFlow adalah pustaka perangkat lunak open-source yang sering digunakan untuk pemrograman dataflow dan diferensial dalam berbagai aplikasi.

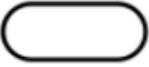
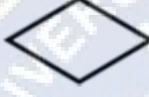
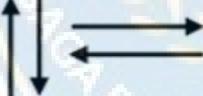
7. Klasifikasi

Klasifikasi merupakan metode yang digunakan untuk mengidentifikasi pola-pola yang dapat menjelaskan atau membedakan antara konsep atau kelas data. Tujuannya adalah untuk memprediksi kelas dari suatu objek yang belum diketahui identitasnya. Sebagai salah satu teknik dalam penambangan data, klasifikasi memasang kelas ke dalam set data tertentu, memungkinkan prediksi dan analisis yang lebih tepat (Syukron et al., 2023).

Klasifikasi adalah proses pengorganisasian objek ke dalam kelompok-kelompok yang anggotanya serupa dengan cara yang sama dan bagian dari pengenalan pola. Dua jenis klasifikasi yaitu klasifikasi supervised dan klasifikasi unsupervised (Wakhidah, n.d.).

8. *TensorFlow*

Table 1. Fungsi *TensorFlow* (DULI, 2024)

Simbol	Nama	Fungsi
	Terminal	Digunakan untuk memulai atau mengakhiri program.
	Inpu/Output	Digunakan untuk menyatakan input atau output tanpa melihat jenisnya.
	Manual Opertion	Digunakan untuk menunjukkan pengolahan yang tidak dilakukan oleh komputer.
	Decision	Digunakan untuk memilih proses yang akan dilakukan berdasarkan kondisi tertentu.
	Processing	Digunakan untuk menunjukkan pengolahan data yang dilakukan oleh komputer.
	Disk Storage	Digunakan untuk menyatakan masukan dan keluaran yang berasal dari disk.
	Flow Direction Symbol/Connecting line	Berfungsi untuk menghubungkan simbol yang satu dengan yang lainnya, menyatakan arus suatu proses.

TensorFlow, atau yang sering disebut diagram alir, adalah jenis diagram yang menggambarkan algoritma atau langkah-langkah instruksi yang berurutan dalam suatu sistem. Seorang analis sistem menggunakan *TensorFlow* sebagai dokumen pendukung untuk menjelaskan gambaran logis dari sebuah sistem yang akan dikembangkan kepada programmer. Dengan demikian, *TensorFlow* dapat membantu dalam memberikan solusi terhadap masalah yang mungkin muncul selama pembangunan sistem. Pada intinya, *TensorFlow* digambarkan menggunakan simbol-simbol, di mana setiap simbol mewakili proses tertentu. Proses-proses tersebut dihubungkan dengan garis penghubung untuk menunjukkan urutan langkah-langkahnya (Rosaly & Prasetyo, 2019).

B. Penelitian Terkait

1. “Pembelajaran Mendalam Pengklasifikasi Ekspresi Wajah Manusia dengan Model Arsitektur Xception pada Metode Convolutional Neural Network” penelitian yang dilakukan Purnawarman Musa, Wahid Khairul Anam, Saiful Bahri Musa, Witari Aryunani, Remi Senjaya, dan Puji Sularsih1 (2023).

Dari hasil penelitian yang dilakukan menggunakan model arsitektur Xception dan Convolutional Neural Networks (*CNN*) untuk klasifikasi ekspresi wajah manusia, ditemukan bahwa akurasi model mencapai 66% dengan presisi rerata 76%, *recall* rerata 65%, dan skor F1 sebesar 63%. Meskipun terdapat kesulitan dalam mengidentifikasi klasifikasi ekspresi jijik karena pengaruh konteks budaya dan keterbatasan data, penelitian merekomendasikan penggunaan intensitas cahaya yang optimal dan penambahan dataset kategori jijik untuk meningkatkan akurasi deteksi ekspresi wajah di masa depan. Metode ini diharapkan dapat memberikan manfaat dalam berbagai bidang seperti interaksi manusia-komputer, pengembangan robot, sistem AI yang lebih cerdas, dan bidang keamanan, serta menunjukkan potensi untuk pengembangan teknologi yang lebih canggih dalam pengenalan ekspresi wajah (Musa et al., 2023).

2. “Pengenalan Ekspresi Wajah Secara Real-Time Menggunakan Metode SSD *MobileNet* Berbasis Android” penelitian yang dilakukan Muhammad Rifqi Daffa Ulhaq, Diash Firdaus, dan Muammar Alfien Zaidan (2023).

Penelitian ini bertujuan untuk mengembangkan metode deteksi ekspresi wajah secara real-time menggunakan pendekatan SSD *MobileNet* berbasis Android. Metode ini mampu mendeteksi ekspresi wajah dengan akurasi tinggi dan kecepatan pemrosesan yang baik, dengan akurasi 0,51 detik dan kecepatan deteksi 31 frame per-detik. Hasil evaluasi menunjukkan bahwa model memiliki akurasi rata-rata 51,2% dan kecepatan pemrosesan 30 fps. Dengan peningkatan jumlah data latih, model ini dapat ditingkatkan lebih lanjut. Metode ini memiliki potensi sebagai solusi efektif untuk pengenalan

ekspresi wajah dalam berbagai aplikasi dan memiliki potensi aplikasi luas dalam berbagai domain (Daffa Ulhaq et al., 2023).

3. “Perancangan Deteksi Emosi Manusia berdasarkan Ekspresi Wajah Menggunakan Algoritma VGG16” penelitian yang dilakukan Dimas Setiawan, Suprih Widodo, Taufik Ridwan, dan Rifqi Ambari (2022).

Penelitian ini berhasil mengimplementasikan deteksi emosi berdasarkan ekspresi wajah menggunakan model VGG 16 dengan tambahan layer dan data training dari dataset FER2013, serta memanfaatkan metode pengenalan wajah menggunakan *MTCNN* untuk hasil yang optimal. Hasil penelitian menunjukkan bahwa model yang dikembangkan siap untuk deployment dengan akurasi pengenalan wajah mencapai 100%. Referensi yang digunakan dalam penelitian mencakup berbagai penelitian terkait pengenalan ekspresi wajah dan penggunaan *Convolutional Neural Networks (CNN)*, serta penerapan *Machine Learning* dalam memecahkan masalah deteksi emosi yang kompleks. Metode penelitian juga melibatkan tahapan *Problem Scoping*, *Data Acquisition*, *Data Exploration*, *Modelling*, *Evaluation*, dan *Deployment* menggunakan framework *AI Project Life Cycle* (Widodo et al., 2022).

4. “Penerapan *Convolutional Neural Network Deep learning* dalam Pendeteksian Citra Biji Jagung Kering” penelitian yang dilakukan Arum TiaraSari, dan Emy Haryatmi(2021).

Penelitian ini menyimpulkan bahwa penerapan *Convolutional Neural Network (CNN)* dalam pendeteksian citra biji jagung kering menghasilkan tingkat akurasi sebesar 90,296% . Hasil evaluasi dari pengujian data testing menunjukkan bahwa sistem mampu mendeteksi biji jagung kering dengan baik dari berbagai sisi dan bentuk ukuran yang berasal dari citra kamera smartphone (TiaraSari & Haryatmi, 2021).

5. “Optimasi Akurasi Metode *Convolutional Neural Network* untuk Identifikasi Jenis Sampah” penelitian yang dilakukan Rima Dias Ramadhani, Afandi Nur Aziz Thohari, Condro Kartiko, Apri Junaidi, Tri Ginanjar Laksana, dan Novanda Alim Setya Nugraha (2020).

Penelitian ini menyimpulkan bahwa Penelitian ini menggunakan *Convolutional Neural Network (CNN)* untuk mengidentifikasi sampah organik dan anorganik dalam citra. Dengan menambahkan hyperparameter seperti dropout, padding, dan stride, akurasi model meningkat dari 67,6% menjadi 91,2%. Optimasi pada arsitektur *CNN* diperlukan untuk mendapatkan hasil yang lebih akurat dalam identifikasi objek dari citra. Penelitian ini membahas tentang proses persiapan, pengkodean, pengujian aplikasi, dan hasil serta pembahasan terkait penggunaan *Convolutional Neural Networks (CNN)* untuk klasifikasi sampah organik dan anorganik. Proses persiapan meliputi koleksi data dari sumber dataset, preprocessing data, dan perancangan aplikasi. Pengkodean dilakukan dengan implementasi algoritma *CNN* menggunakan library Tensor Flow dan *Keras*. Pengujian aplikasi dilakukan menggunakan perangkat smartphone Android dengan alat pengujian Espresso. Hasil pelatihan model *CNN* menunjukkan peningkatan akurasi setelah dilakukan optimasi hyperparameter (Rima Dias Ramadhani et al., 2021).

6. “Convolution Neural Network (*CNN*) Untuk Pengklasifikasian Citra Makanan Tradisional” penelitian yang dilakukan Akhmad Rohim dan Yuita Arum Sari, (2019).

Penelitian ini menyimpulkan bahwa dalam pengembangan model *Convolutional Neural Network* untuk klasifikasi citra makanan tradisional, digunakan arsitektur yang terdiri dari 4 layer konvolusi, 4 layer maxpooling, dan 2 layer Fully Connected. Arsitektur ini dipilih karena menghasilkan nilai loss terendah, yaitu 0.000044, pada epoch ke-15 selama proses pembelajaran. Model ini juga mencapai presisi 73%, *recall* 69%, dan Fscore 69%. Penelitian ini menunjukkan bahwa hasil klasifikasi citra makanan dapat dipengaruhi oleh sudut pandang objek dalam citra. Oleh karena itu, untuk penelitian selanjutnya, disarankan untuk meningkatkan variasi data dengan mengambil citra makanan dari berbagai sudut pandang. Diharapkan peningkatan variasi sudut pandang ini dapat meningkatkan presisi dan *recall* model (Rohim et al., 2019).

Table 2. Persamaan dan Perbedaan penelitian

Judul	Persamaan	Perbedaan
<p>“Pembelajaran Mendalam Pengklasifikasi Ekspresi Wajah Manusia dengan Model Arsitektur <i>Xception</i> pada Metode <i>Convolutional Neural Network</i>” penelitian yang dilakukan Purnawarman Musa, Wahid Khairul Anam, Saiful Bahri Musa, Witari Aryunani, Remi Senjaya, dan Puji Sularsih1 (2023).</p>	<ul style="list-style-type: none"> • Kedua penelitian menggunakan metode <i>CNN</i>. • Fokus pada pengenalan ekspresi wajah manusia. 	<ul style="list-style-type: none"> • Model yang digunakan berbeda: <i>Xception</i> vs <i>MobileNet</i>. • Akurasi dan metrik evaluasi yang berbeda.
<p>“Pengenalan Ekspresi Wajah Secara <i>Real-Time</i> Menggunakan Metode SSD <i>MobileNet</i> Berbasis Android” penelitian yang dilakukan Muhammad Rifqi Daffa Ulhaq, Diash Firdaus, dan Muammar Alfien Zaidan (2023).</p>	<ul style="list-style-type: none"> • Keduanya menggunakan arsitektur <i>MobileNet</i>. • Fokus pada deteksi ekspresi wajah. 	<ul style="list-style-type: none"> • Implementasi berbasis Android. • Metrik kinerja berbeda: kecepatan pemrosesan dan akurasi.
<p>“Perancangan Emosi Deteksi Manusia</p>	<ul style="list-style-type: none"> • Keduanya menggunakan 	<ul style="list-style-type: none"> • Model yang digunakan berbeda:

berdasarkan Ekspresi metode *CNN* VGG16 vs
 Wajah Menggunakan untuk pengenalan *MobileNet*.
 Algoritma VGG16” emosi manusia. • Dataset yang
 penelitian yang dilakukan • Fokus pada digunakan berbeda.
 Dimas Setiawan,Suprih pengenalan
 Widodo, Taufik Ridwan, ekspresi wajah.
 dan Rifqi Ambari (2022).

“Penerapan • Sama-sama • Domain aplikasi
Convolutional Neural menggunakan berbeda:
Network Deep learning *CNN*. pengenalan wajah
 dalam Pendeteksian Citra vs deteksi citra biji
 Biji Jagung Kering” jagung.
 penelitian yang dilakukan • Tidak fokus pada
 Arum TiaraSari, dan Emy pengenalan
 Haryatmi(2021). ekspresi wajah.

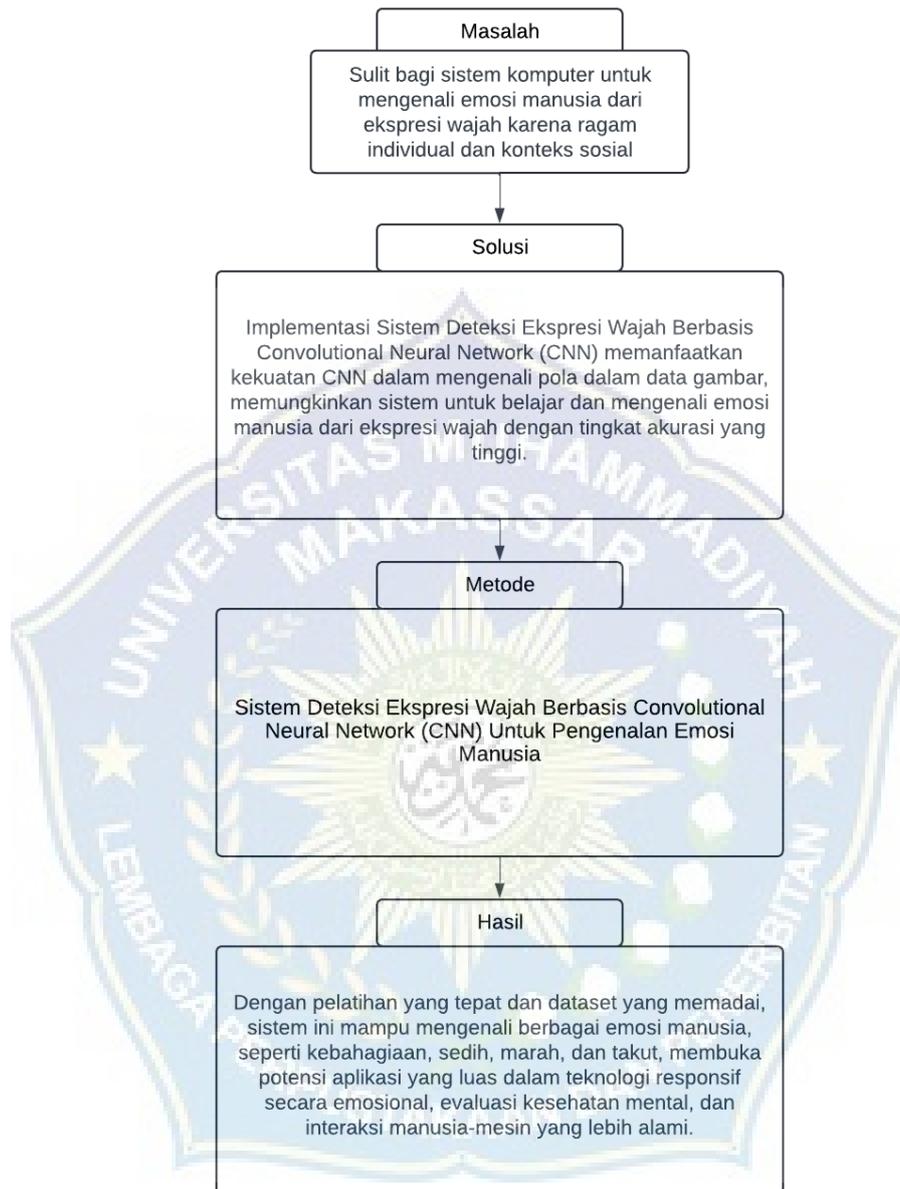
“Optimasi Akurasi • Sama-sama • Domain aplikasi
 Metode *Convolutional* menggunakan berbeda:
Neural Network untuk *CNN*. pengenalan wajah
 Identifikasi Jenis vs identifikasi
 Sampah” penelitian yang sampah.
 dilakukan Rima Dias • Fokus pada
 Ramadhani, Afandi Nur optimasi
 Aziz Thohari, Condro hyperparameter.
 Kartiko, Apri Junaidi, Tri
 Ginanjar Laksana, dan
 Novanda Alim Setya
 Nugraha (2020).

“*Convolution Neural Network (CNN) Untuk Pengklasifikasian Citra Makanan Tradisional*” penelitian yang dilakukan Akhmad Rohim dan Yuita Arum Sari, (2019).

- Sama-sama menggunakan *CNN*.
- Domain aplikasi berbeda: pengenalan wajah vs klasifikasi makanan.



C. Kerangka Pikir



Gambar 6. Kerangka Pikir

BAB III

METODE PENELITIAN

A. Tempat dan Waktu Penelitian

Penelitian yang berfokus pada eksplorasi ekspresi wajah seseorang memerlukan keberagaman tempat untuk mengumpulkan sampel yang representatif. Tempat penelitian ini meliputi berbagai konteks, mulai dari ruang publik. Selain itu, ruang privat seperti rumah tinggal juga menjadi tempat relevan untuk menangkap ekspresi yang lebih intim dan alami. Dengan demikian, keberagaman tempat penelitian memungkinkan peneliti untuk memperoleh gambaran yang lebih lengkap dan mendalam tentang berbagai macam ekspresi manusia dalam berbagai konteks kehidupan sehari-hari.

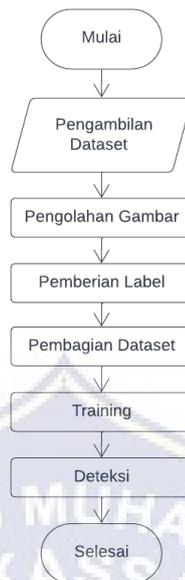
B. Bahan dan Alat Penelitian

Alat yang diperlukan dalam penelitian ini adalah sebagai berikut:

1. Kebutuhan Hardware (Perangkat Keras)
 - a. Smartphone Realme 10 Pro+ 5G
 - b. Laptop Lenovo Ideapad Gaming 3
2. Kebutuhan Software
 - a. OS Windows 11
 - b. *Python*
 - c. Visual Studio Code
 - d. Google Colaboratory

C. Perancangan Sistem

TensorFlow merupakan representasi visual dari urutan langkah-langkah atau proses dalam sistem pendeteksian ekspresi manusia menggunakan simbol-simbol grafis untuk menggambarkan algoritma *CNN*. Berikut adalah *TensorFlow* perancangan sistem dari pendeteksi ekspresi manusia menggunakan *CNN*.



Gambar 8. Perancangan Sistem

1. Pengambilan dataset

Data diambil menggunakan kamera smartphone dari beberapa subjek manusia yang mengekspresikan lima jenis emosi: netral, senang, sedih, marah, dan takut. Dataset ini akan digunakan sebagai contoh dalam melatih sistem pendeteksi ekspresi yang sedang dikembangkan.

2. Pengolahan gambar

Dalam pengolahan data diperlukan preprocessing data yang dimana Preprocessing merupakan tahap krusial sebelum memasukkan citra ke dalam model atau algoritma, yang bertujuan untuk menyiapkan dan meningkatkan kualitas atau kesesuaian citra sesuai dengan kebutuhan.

3. Pemberian label

Dalam langkah ini, dataset telah diproses dengan pengolahan gambar yang diberi label menggunakan platform *Roboflow*. *Roboflow* bertujuan untuk menambahkan kotak pembatas pada setiap objek yang terdapat dalam gambar, yang nantinya akan digunakan untuk melatih model.

4. Pembagian dataset

Dataset yang telah diberi label akan dibagi menjadi tiga bagian yaitu data training, data validasi, dan data testing. Data training digunakan untuk melatih model deteksi, sementara data validasi digunakan untuk mengevaluasi performa model selama proses pelatihan. Data testing kemudian digunakan untuk menguji performa model dan mengukur akurasi dengan menggunakan data yang belum pernah dilihat sebelumnya.

5. Training

Pada langkah ini, data yang telah dibagi menjadi data latih dan data uji akan diakses. Data yang tersedia dalam bentuk API di *Roboflow* akan diambil melalui lingkungan pengembangan seperti *VSCode*. Selanjutnya, model *CNN* akan diuji dengan data tersebut, dan akurasi akan digunakan sebagai parameter evaluasi utama.

Jika hasil pengujian menunjukkan bahwa akurasi model masih rendah, langkah selanjutnya adalah melakukan pelatihan ulang model. Proses pelatihan ulang ini melibatkan kembali melatih model *CNN* dengan menggunakan data latih, dengan tujuan untuk meningkatkan akurasi dan kemampuan model dalam mengenali ekspresi wajah.

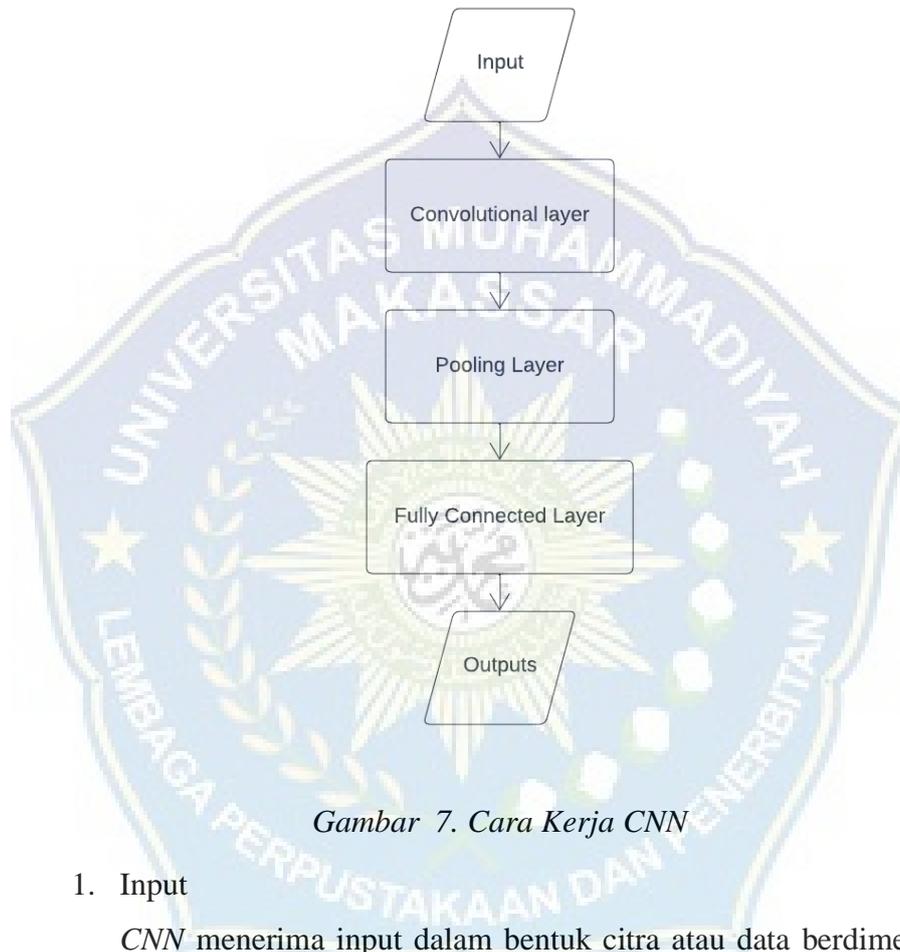
Dengan demikian, siklus pengembangan model melibatkan iterasi berulang antara melatih model, menguji model, dan jika perlu, melakukan pelatihan ulang untuk terus meningkatkan kualitas dan performa model hingga mencapai tingkat akurasi yang diinginkan.

6. Deteksi

Proses pendeteksian dimulai dengan memasukkan gambar yang telah dikumpulkan ke dalam sistem. Gambar tersebut kemudian dianalisis menggunakan model *CNN* untuk mengidentifikasi objek. Hasil dari proses ini mencakup akurasi, yang menunjukkan seberapa efektif model dalam mengenali objek, serta *ROI (Region of Interest)* untuk menentukan area yang relevan untuk deteksi.

D. Cara kerja CNN

Berikut adalah *TensorFlow* cara kerja CNN secara umum yang meliputi 3 jenis lapisan utama, yaitu *Convolutional layer*, *Pooling Layer*, dan *Fully Connected Layer*.



Gambar 7. Cara Kerja CNN

1. Input

CNN menerima input dalam bentuk citra atau data berdimensi tinggi lainnya. Sebagai contoh, citra diwakili sebagai matriks piksel, dengan setiap piksel memiliki nilai yang menunjukkan intensitas warna atau kecerahan.

2. Convolutional Layer

Lapisan ini terdiri dari sejumlah filter atau kernel dengan ukuran kecil yang diterapkan ke seluruh input. Setiap filter berinteraksi dengan input melalui operasi konvolusi, di mana filter tersebut digeser di atas input dan melakukan perkalian titik dengan bagian yang sesuai dari input.

Hasil dari operasi ini menghasilkan aktivasi yang menyoroti fitur-fitur spesifik dalam input. Filter-filter ini belajar untuk mengidentifikasi berbagai fitur pada gambar, seperti tepi, tekstur, atau pola-pola penting lainnya.

3. *Pooling Layer*

Setelah lapisan konvolusi, biasanya diikuti oleh lapisan pooling, yang bertujuan untuk mengurangi dimensi spasial dari representasi gambar serta jumlah parameter dalam jaringan. Hal ini dilakukan untuk mengurangi kebutuhan komputasi dan mencegah overfitting. Salah satu teknik pooling yang umum adalah max pooling, di mana nilai maksimum dari setiap subregion dari input diambil dan dijadikan sebagai representasi subregion tersebut (Rafiedhia Pramutighna & Hermawan, 2023). Ini membantu mempertahankan fitur-fitur dominan dari gambar sambil mengurangi resolusi spasial, sehingga meningkatkan efisiensi komputasi.

4. *Fully Connected Layer*

Setelah melalui beberapa lapisan konvolusi dan pooling, representasi gambar diubah menjadi vektor dan diteruskan ke lapisan-lapisan yang sepenuhnya terhubung (Rafiedhia Pramutighna & Hermawan, 2023). Lapisan-lapisan ini beroperasi seperti jaringan saraf biasa (*feedforward neural network*), di mana setiap neuron di satu lapisan terhubung dengan semua neuron di lapisan berikutnya. Koneksi ini memungkinkan jaringan untuk mempelajari hubungan yang semakin kompleks antara fitur-fitur yang telah diekstraksi sebelumnya. Dengan demikian, informasi yang dihasilkan dari lapisan-lapisan sebelumnya dapat diolah lebih lanjut untuk melakukan tugas-tugas yang lebih kompleks, seperti klasifikasi objek dalam gambar.

5. *Outputs*

Pada tahap akhir, output dari lapisan-lapisan sepenuhnya terhubung diambil dan diteruskan ke lapisan output. Struktur lapisan output dapat bervariasi tergantung pada jenis tugas yang dihadapi. Dengan demikian,

proses ini menetapkan label atau kategori kepada input yang telah diproses melalui jaringan *CNN*, memungkinkan model untuk melakukan klasifikasi atau tugas lainnya sesuai dengan tujuan yang ditetapkan.

E. Pengujian sistem

Sistem pengujian empiris dalam penelitian tentang sistem deteksi ekspresi wajah berbasis *CNN* untuk pengenalan emosi manusia melibatkan serangkaian langkah dan metode untuk menguji kinerja sistem yang dikembangkan dalam penelitian. Berikut adalah beberapa langkah umum yang biasanya dilakukan dalam pengujian empiris:

1. Pemilihan Dataset

Pemilihan dataset yang cocok untuk pengenalan ekspresi wajah adalah langkah awal yang krusial. Dataset tersebut harus mencakup berbagai ekspresi wajah manusia dalam kondisi yang beragam, seperti cahaya yang berbeda, latar belakang yang berbeda, sudut pandang yang berbeda, dan etnisitas yang berbeda.

2. Preprocessing Data

Data wajah dari dataset harus dipreproses sebelum dimasukkan ke dalam model *CNN*. Langkah ini meliputi normalisasi, resizing, dan augmentasi data jika diperlukan.

3. Pembagian Data

Dataset harus dibagi menjadi tiga subset: data pelatihan, data validasi, dan data pengujian. Data pelatihan digunakan untuk melatih model, data validasi digunakan untuk menyesuaikan parameter model dan memperkirakan kinerjanya selama pelatihan, sementara data pengujian digunakan untuk menguji kinerja model secara independen setelah pelatihan selesai.

4. Pengaturan Model *CNN*

Pembangunan model *CNN* yang tepat adalah langkah penting berikutnya. Ini melibatkan pemilihan arsitektur *CNN* yang sesuai, seperti ResNet,

VGG, atau arsitektur khusus lainnya yang telah terbukti efektif dalam tugas pengenalan emosi dari wajah.

5. Pelatihan Model

Model *CNN* kemudian dilatih menggunakan data pelatihan yang dipersiapkan sebelumnya. Selama pelatihan, parameter model diperbarui secara iteratif untuk mengurangi kesalahan prediksi.

6. Validasi Model

Setelah pelatihan, model divalidasi menggunakan data validasi yang terpisah. Ini membantu dalam menyesuaikan parameter model dan mencegah overfitting.

7. Pengujian Model

Setelah model dianggap cukup valid, kinerjanya diuji menggunakan data pengujian yang terpisah.

8. Analisis Hasil

Hasil pengujian dianalisis untuk mengevaluasi kekuatan dan kelemahan model. Ini dapat melibatkan analisis kesalahan, visualisasi prediksi, dan perbandingan kinerja model dengan penelitian sebelumnya atau model lain yang ada.

9. Kesimpulan

Berdasarkan hasil pengujian, kesimpulan ditarik mengenai keefektifan sistem deteksi ekspresi wajah berbasis *CNN* untuk pengenalan emosi manusia. Hasilnya juga dapat dibandingkan dengan literatur sebelumnya untuk mengidentifikasi kontribusi unik penelitian tersebut.

F. Teknik dan Analisis Data

Analisis data adalah proses sistematis yang melibatkan pengumpulan, pengelompokan, dan seleksi data dari berbagai sumber seperti wawancara, catatan lapangan, dan dokumentasi. Proses ini mencakup pengorganisasian data ke dalam kategori, merangkumnya ke dalam unit-unit, melakukan sintesis, mengidentifikasi pola, menentukan data yang relevan untuk dianalisis, dan menarik kesimpulan yang jelas. Terdapat tiga teknik utama dalam analisis data:

1. Reduksi data

Mereduksi data berarti merangkum dan memilih informasi yang pokok serta memfokuskan pada hal-hal yang penting. Tujuannya adalah untuk mengolah data mentah menjadi lebih bermakna, memberikan gambaran yang jelas, dan mempermudah pengumpulan data selanjutnya.

2. Penyajian data

Penyajian data adalah tata cara penyusunan informasi secara sistematis yang memungkinkan untuk menarik kesimpulan dan mengambil tindakan. Ini melibatkan pengaturan informasi secara terstruktur untuk memfasilitasi analisis lebih lanjut.

3. Penarikan kesimpulan

Penarikan kesimpulan merupakan tahap akhir dalam analisis data, di mana kesimpulan awal dibuat berdasarkan hasil analisis sementara. Kesimpulan ini dapat berubah seiring dengan ditemukannya bukti baru pada tahap pengumpulan data selanjutnya. Tujuannya adalah untuk memahami makna, pola, dan keteraturan dari data yang telah dianalisis.

BAB VI HASIL DAN PEMBAHASAN

A. Pembuatan Model

1. Pengambilan Dataset

Dataset gambar diambil menggunakan kamera HP dan terdiri dari 800 gambar ekspresi wajah. Dataset ini mencakup berbagai ekspresi, termasuk senang, sedih, marah, dan takut.

Table 3. Jumlah Dataset

Tempat pengambilan data	Objek Data Yang di ambil	Jumlah Data Gambar
Universitas Muhammadiyah Makassar	Ekspresi Senang	200
	Ekspresi Sedih	200
	Ekspresi Marah	200
	Ekspresi Takut	200
	Jumlah Data Gambar	800

a. Gambar ekspresi senang



Gambar 8. Contoh Ekspresi Senang

b. Gambar ekspresi sedih



Gambar 9. Contoh Ekspresi Sedih

c. Gambar ekspresi marah



Gambar 10. Contoh Ekspresi Marah

d. Gambar ekspresi takut

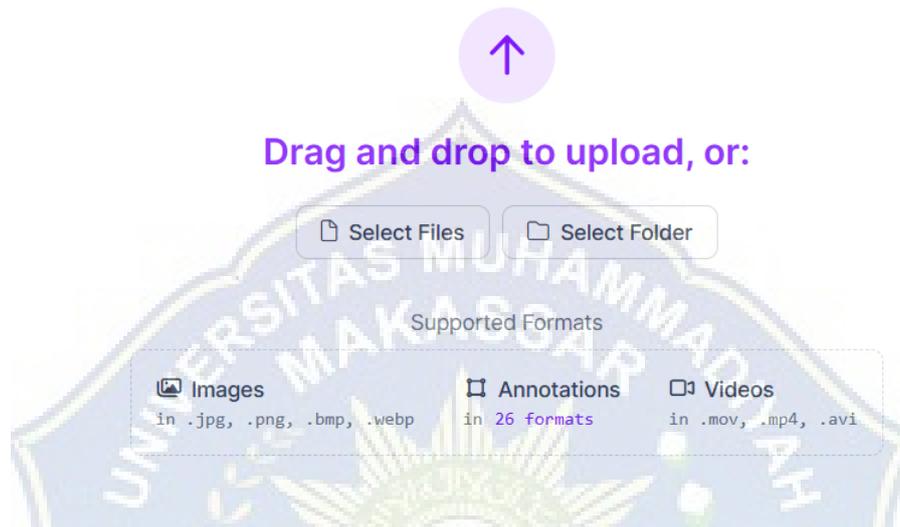


Gambar 11. Contoh Ekspresi Takut

2. Pelabelan

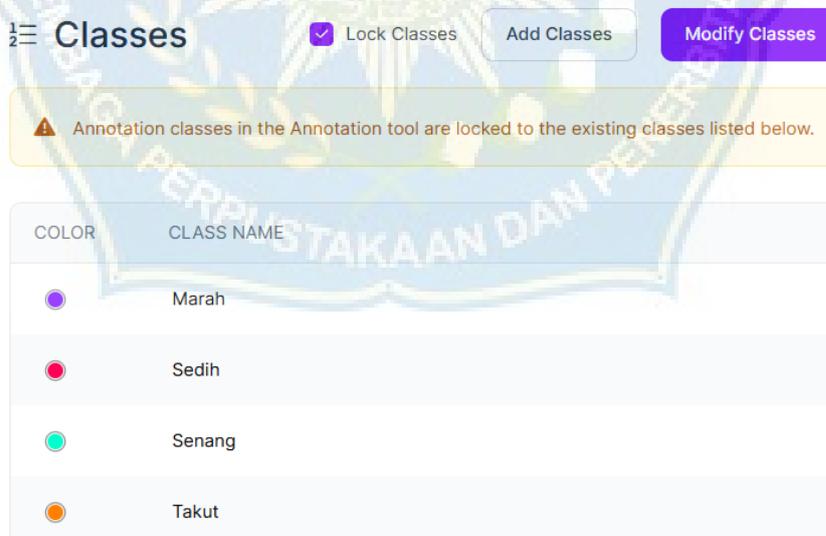
a. Upload dataset

Gambar yang telah dikumpulkan sebelumnya akan diunggah ke *Roboflow* sebelum proses pelabelan dilakukan.



Gambar 12. Upload Dataset

b. Pembuatan Class Dataset



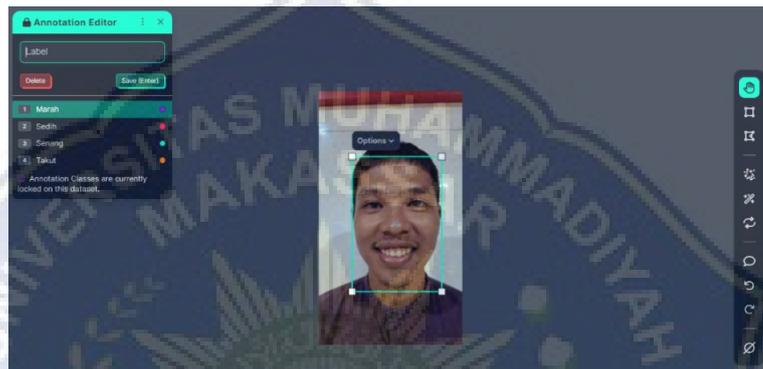
Gambar 13. Class Dataset

Tujuan pembuatan kelas ini adalah untuk mempermudah penentuan dataset gambar yang telah diberi bounding box, sehingga secara otomatis akan

muncul lima kelas yang telah dibuat. Dengan demikian, dataset akan masuk ke dalam kategori yang sesuai.

c. Proses pelabelan dan pembagian kategori gambar

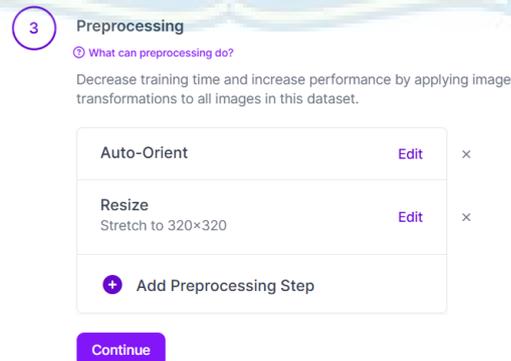
Proses pelabelan gambar menggunakan alat *Roboflow* dilakukan dengan menambahkan bounding box atau frame pada objek dalam gambar. Setelah bounding box ditambahkan, kelas dengan kategori senang, sedih, marah, dan takut akan muncul secara otomatis.



Gambar 14. Pelabelan Dataset

3. Preprocessing Gambar

Proses pengolahan gambar melibatkan pengeditan gambar menggunakan *Roboflow*, termasuk penyesuaian warna, bentuk, dan ukuran, serta penggunaan teknik seperti *Preprocessing* dan *Augmentation*. Mengubah ukuran gambar (resize) menjadi 320 x 320 bertujuan untuk mengurangi beban kerja GPU saat melatih model.



Gambar 15. Preprocessing Gambar

B. Training Googlecolab

```

Epoch 138/150
-----
| ETA: 0s - loss: 7.9123 - bboxes_output_loss: 7.6653 - class_output_loss: 0.2471 - bboxes_output_mse: 7.6653 - class_output_accuracy: 0.9453
Epoch 139/150
-----
| val_loss: 81.8458 - val_bboxes_output_loss: 80.8598 - val_class_output_loss: 0.9861 - val_bboxes_output_mse: 80.8598 - val_class_output_accuracy: 0.5297
Epoch 140/150
-----
| val_loss: 81.8458 - val_bboxes_output_loss: 80.8598 - val_class_output_loss: 0.9861 - val_bboxes_output_mse: 80.8598 - val_class_output_accuracy: 0.5297
Epoch 141/150
-----
| ETA: 0s - loss: 7.2118 - bboxes_output_loss: 6.9652 - class_output_loss: 0.2458 - bboxes_output_mse: 6.9652 - class_output_accuracy: 0.9207
Epoch 142/150
-----
| val_loss: 63.1838 - val_bboxes_output_loss: 62.4313 - val_class_output_loss: 0.7525 - val_bboxes_output_mse: 62.4313 - val_class_output_accuracy: 0.6787
Epoch 143/150
-----
| val_loss: 63.1838 - val_bboxes_output_loss: 62.4313 - val_class_output_loss: 0.7525 - val_bboxes_output_mse: 62.4313 - val_class_output_accuracy: 0.6787
Epoch 144/150
-----
| ETA: 0s - loss: 5.2885 - bboxes_output_loss: 5.0327 - class_output_loss: 0.2558 - bboxes_output_mse: 5.0327 - class_output_accuracy: 0.9404
Epoch 145/150
-----
| val_loss: 69.1459 - val_bboxes_output_loss: 68.4715 - val_class_output_loss: 0.6744 - val_bboxes_output_mse: 68.4715 - val_class_output_accuracy: 0.6991
Epoch 146/150
-----
| val_loss: 69.1459 - val_bboxes_output_loss: 68.4715 - val_class_output_loss: 0.6744 - val_bboxes_output_mse: 68.4715 - val_class_output_accuracy: 0.6991
Epoch 147/150
-----
| ETA: 0s - loss: 5.1428 - bboxes_output_loss: 4.9489 - class_output_loss: 0.1939 - bboxes_output_mse: 4.9489 - class_output_accuracy: 0.9563
Epoch 148/150
-----
| val_loss: 66.2461 - val_bboxes_output_loss: 65.4317 - val_class_output_loss: 0.8084 - val_bboxes_output_mse: 65.4317 - val_class_output_accuracy: 0.6884
Epoch 149/150
-----
| val_loss: 66.2461 - val_bboxes_output_loss: 65.4317 - val_class_output_loss: 0.8084 - val_bboxes_output_mse: 65.4317 - val_class_output_accuracy: 0.6884
Epoch 150/150
-----
| ETA: 0s - loss: 4.1725 - bboxes_output_loss: 3.9797 - class_output_loss: 0.1928 - bboxes_output_mse: 3.9797 - class_output_accuracy: 0.9783
Epoch 151/150
-----
| val_loss: 77.1646 - val_bboxes_output_loss: 76.1554 - val_class_output_loss: 1.0113 - val_bboxes_output_mse: 76.1554 - val_class_output_accuracy: 0.6113
Epoch 152/150
-----
| val_loss: 77.1646 - val_bboxes_output_loss: 76.1554 - val_class_output_loss: 1.0113 - val_bboxes_output_mse: 76.1554 - val_class_output_accuracy: 0.6113
Epoch 153/150
-----
| ETA: 0s - loss: 5.5481 - bboxes_output_loss: 5.2791 - class_output_loss: 0.2690 - bboxes_output_mse: 5.2791 - class_output_accuracy: 0.9758
Epoch 154/150
-----
| val_loss: 62.4313 - val_bboxes_output_loss: 61.9356 - val_class_output_loss: 0.4918 - val_bboxes_output_mse: 61.9356 - val_class_output_accuracy: 0.7819
Epoch 155/150
-----
| val_loss: 62.4313 - val_bboxes_output_loss: 61.9356 - val_class_output_loss: 0.4918 - val_bboxes_output_mse: 61.9356 - val_class_output_accuracy: 0.7819
Epoch 156/150
-----
| ETA: 0s - loss: 5.4585 - bboxes_output_loss: 5.2791 - class_output_loss: 0.1794 - bboxes_output_mse: 5.2791 - class_output_accuracy: 0.9766
Epoch 157/150
-----
| val_loss: 72.8333 - val_bboxes_output_loss: 72.1786 - val_class_output_loss: 0.6587 - val_bboxes_output_mse: 72.1786 - val_class_output_accuracy: 0.6787
Epoch 158/150
-----
| val_loss: 72.8333 - val_bboxes_output_loss: 72.1786 - val_class_output_loss: 0.6587 - val_bboxes_output_mse: 72.1786 - val_class_output_accuracy: 0.6787
Epoch 159/150
-----
| ETA: 0s - loss: 4.8538 - bboxes_output_loss: 4.6822 - class_output_loss: 0.1716 - bboxes_output_mse: 4.6822 - class_output_accuracy: 0.9797
Epoch 160/150
-----
| val_loss: 66.2611 - val_bboxes_output_loss: 65.6838 - val_class_output_loss: 0.5793 - val_bboxes_output_mse: 65.6838 - val_class_output_accuracy: 0.5793
Epoch 161/150
-----
| val_loss: 66.2611 - val_bboxes_output_loss: 65.6838 - val_class_output_loss: 0.5793 - val_bboxes_output_mse: 65.6838 - val_class_output_accuracy: 0.5793
Epoch 162/150
-----
| ETA: 0s - loss: 4.7354 - bboxes_output_loss: 4.6022 - class_output_loss: 0.1331 - bboxes_output_mse: 4.6022 - class_output_accuracy: 0.9891
Epoch 163/150
-----
| val_loss: 69.8722 - val_bboxes_output_loss: 68.5458 - val_class_output_loss: 0.5283 - val_bboxes_output_mse: 68.5458 - val_class_output_accuracy: 0.5283
Epoch 164/150
-----
| val_loss: 69.8722 - val_bboxes_output_loss: 68.5458 - val_class_output_loss: 0.5283 - val_bboxes_output_mse: 68.5458 - val_class_output_accuracy: 0.5283
Epoch 165/150
-----
| ETA: 0s - loss: 3.9347 - bboxes_output_loss: 3.7683 - class_output_loss: 0.1664 - bboxes_output_mse: 3.7683 - class_output_accuracy: 0.9766
Epoch 166/150
-----
| val_loss: 72.8617 - val_bboxes_output_loss: 71.2167 - val_class_output_loss: 0.8218 - val_bboxes_output_mse: 71.2167 - val_class_output_accuracy: 0.8218
Epoch 167/150
-----
| val_loss: 72.8617 - val_bboxes_output_loss: 71.2167 - val_class_output_loss: 0.8218 - val_bboxes_output_mse: 71.2167 - val_class_output_accuracy: 0.8218
Epoch 168/150
-----
| ETA: 0s - loss: 3.7884 - bboxes_output_loss: 3.6331 - class_output_loss: 0.1553 - bboxes_output_mse: 3.6331 - class_output_accuracy: 0.9672
Epoch 169/150
-----
| val_loss: 67.6245 - val_bboxes_output_loss: 67.0999 - val_class_output_loss: 0.5246 - val_bboxes_output_mse: 67.0999 - val_class_output_accuracy: 0.5246
Epoch 170/150
-----
| val_loss: 67.6245 - val_bboxes_output_loss: 67.0999 - val_class_output_loss: 0.5246 - val_bboxes_output_mse: 67.0999 - val_class_output_accuracy: 0.5246
Epoch 171/150
-----
| ETA: 0s - loss: 5.4889 - bboxes_output_loss: 5.2849 - class_output_loss: 0.2040 - bboxes_output_mse: 5.2849 - class_output_accuracy: 0.9875
Epoch 172/150
-----
| val_loss: 81.9981 - val_bboxes_output_loss: 81.3691 - val_class_output_loss: 0.6218 - val_bboxes_output_mse: 81.3691 - val_class_output_accuracy: 0.6218
Epoch 173/150
-----
| val_loss: 81.9981 - val_bboxes_output_loss: 81.3691 - val_class_output_loss: 0.6218 - val_bboxes_output_mse: 81.3691 - val_class_output_accuracy: 0.6218

```

Gambar 16. Proses Training

Setelah melakukan percobaan training di Googlecolab dengan menggunakan 80% data pelatihan, dan 20% data pengujian. Dengan menggunakan model *CNN MobileNet* dengan epoch 150 telah di peroleh rata-rata akurasi sebesar 78%. Selanjutnya model di simpan dalam model_checkpoint2.h5 untuk dapat digunakan dalam deteksi ekspresi wajah.

```

5/5 [=====] - 0s 69ms/step
Confusion Matrix:
[[27  1  1  8]
 [ 9 17  1  8]
 [ 2  1 46  1]
 [ 2  1  1 34]]
Classification Report:

```

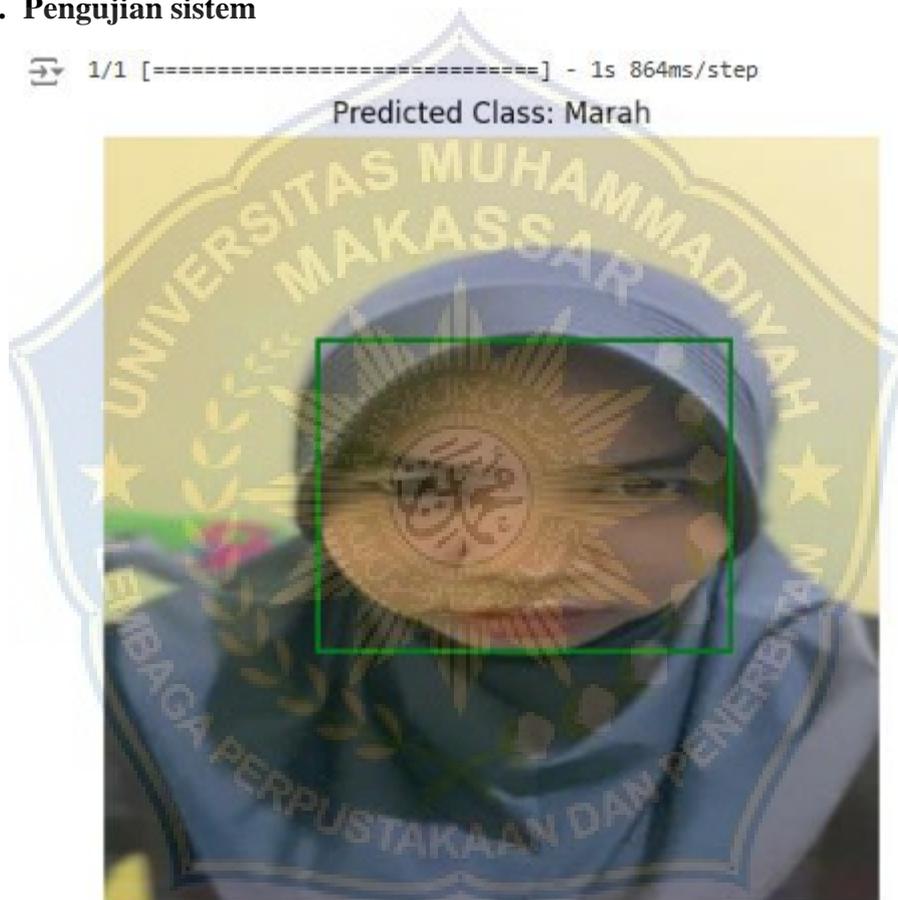
	precision	recall	f1-score	support
Marah	0.68	0.73	0.70	37
Sedih	0.85	0.49	0.62	35
Senang	0.94	0.92	0.93	50
Takut	0.67	0.89	0.76	38
accuracy			0.78	160
macro avg	0.78	0.76	0.75	160
weighted avg	0.79	0.78	0.77	160

Gambar 17. Hasil Dari Training

Confusion matrix menunjukkan distribusi prediksi model untuk setiap kelas, di mana diagonal utama menunjukkan jumlah prediksi yang benar untuk masing-masing kelas. Evaluasi lebih rinci menggunakan *Precision*, *recall*, dan *f1-score* untuk setiap emosi menunjukkan bahwa model memiliki performa terbaik dalam mengklasifikasikan emosi "Senang" dengan *f1-score* 0.93,

diikuti oleh "Takut" dengan *f1-score* 0.76, dan "Marah" dengan *f1-score* 0.70. Namun, model menunjukkan kinerja terburuk dalam mengklasifikasikan emosi "Sedih" dengan *recall* hanya 0.49, yang mengindikasikan bahwa model sering gagal mengidentifikasi emosi ini dengan benar. Secara keseluruhan, model memiliki rata-rata akurasi sebesar 0.78, dengan *Precision* rata-rata (macro avg) 0.78, *recall* rata-rata 0.76, dan *f1-score* rata-rata 0.75.

C. Pengujian sistem



Gambar 18. Gambar dengan prediksi bounding box

Gambar diatas menunjukkan bahwa program dapat memprediksi gambar marah dengan aktual marah, yang menunjukkan prediksi deteksi ekspresi benar.

```
1/1 [=====] - 0s 388ms/step  
class_pred: [1]  
classes_list: ['Marah', 'Sedih', 'Senang', 'Takut']
```



Gambar 19. Hasil Deteksi Gambar

Setelah melakukan percobaan prediksi gambar, selanjutnya program dapat mendeteksi ekspresi marah, senang, takut dan sedih yang di ambil dari model yang tersimpan pada checkpoin pada saat training data. Hasil deteksi tersebut menampilkan gambar uji bersama dengan bounding box yang digambar berdasarkan prediksi, dan menampilkan label kelas hasil prediksi di judul plot. Berikut adalah hasil pengujian deteksi ekspresi.

Table 4. Hasil Pengujian

Tabel Hasil Pengujian

Data Pengujian	Kelas Dataset	Hasil Deteksi
	Marah	
	Senang	
	Takut	
	Sedih	

Tabel di atas menunjukkan hasil pengujian model deteksi emosi, yang terdiri dari tiga kolom: "Data Pengujian," "Kelas Dataset," dan "Hasil Deteksi." Dalam baris pertama, gambar seseorang dengan ekspresi marah berhasil dideteksi dengan benar oleh model sebagai marah. Baris kedua menunjukkan gambar seseorang yang tersenyum, yang juga terdeteksi dengan benar sebagai senang oleh model. Pada baris ketiga, ekspresi takut pada gambar diuji, dan model berhasil mendeteksinya dengan akurat sebagai takut. Terakhir, baris keempat menampilkan gambar seseorang yang sedih, dan model

mendeteksinya dengan tepat sebagai sedih. Secara keseluruhan, tabel ini mengilustrasikan bahwa model mampu mendeteksi ekspresi emosi sesuai dengan label aktual dari dataset.



Gambar 20. Prediksi dan Label Asli

Table 5. Identifikasi Prediksi dan Label Asli

Data	Prediksi	Label Asli
Gambar 1	Marah	Marah
Gambar 2	Marah	Marah
Gambar 3	Senang	Senang
Gambar 4	Senang	Senang
Gambar 5	Takut	Takut
Gambar 6	Marah	Marah
Gambar 7	Sedih	Sedih
Gambar 8	Marah	Takut
Gambar 9	Takut	Takut
Gambar 10	Takut	Takut

Table 6. TP, TN, FP, dan FN

Kelas	True Positif	True Negative	False Positif	False Negative
Marah	3	6	1	0
Sedih	1	9	0	0
Senang	2	8	0	0
Takut	3	6	0	1

Untuk perhitungan *Precision*, *recall*, dan f1- score berdasarkan tabel diatas, dengan menggunakan rumus berikut:

$$\text{Precision} = \frac{TP}{TP+FP}$$

$$\text{Recall} = \frac{TP}{TP+FN}$$

$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Perhitungan setiap kelas:

Marah

$$TP = 3, FP = 1, FN = 0$$

$$\text{Precision} = \frac{3}{3+1} = \frac{3}{4} = 0.75$$

$$\text{Recall} = \frac{3}{3+0} = 1$$

$$\text{F1-score} = 2 \times \frac{0.75 \times 1}{0.75 + 1} = 0.857$$

Sedih

$$TP = 1, FP = 0, FN = 0$$

$$\text{Precision} = \frac{1}{1+0} = 1$$

$$\text{Recall} = \frac{1}{1+0} = 1$$

$$\text{F1-score} = 2 \times \frac{1 \times 1}{1+1} = 1$$

Senang

$$TP = 2, FP = 0, FN = 0$$

$$\text{Precision} = \frac{2}{2+0} = 1$$

$$\text{Recall} = \frac{2}{2+0} = 1$$

$$\text{F1-score} = 2 \times \frac{1 \times 1}{1+1} = 1$$

Takut

TP = 3, FP = 0, FN = 1

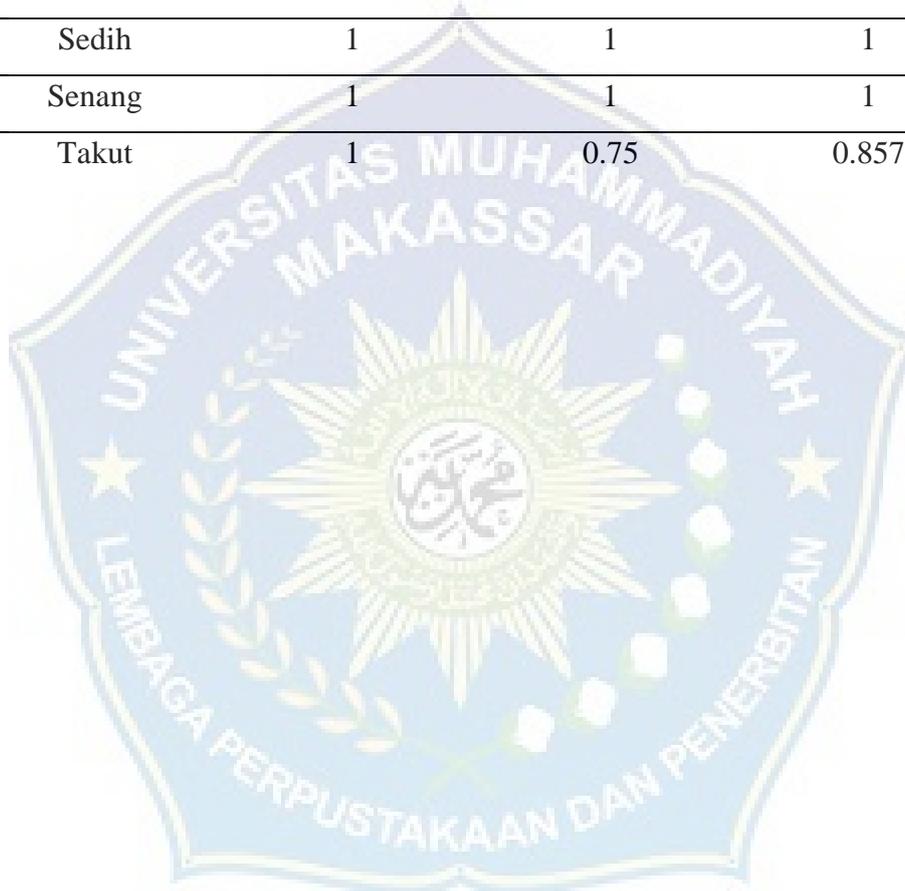
$$\text{Precision} = \frac{3}{3+0} = 1$$

$$\text{Recall} = \frac{3}{3+1} = \frac{3}{4} = 0.75$$

$$\text{F1-score} = 2 \times \frac{1 \times 0.75}{1+0.75} = 0.857$$

Table 7. Hasil Precision, recall, dan f1-score

Kelas	Precision	Recall	F1-score
Marah	0.75	1	0.857
Sedih	1	1	1
Senang	1	1	1
Takut	1	0.75	0.857



BAB V

PENUTUP

A. KESIMPULAN

1. Penerapan metode *CNN* dengan model *MobileNet* pada sistem deteksi ekspresi wajah memungkinkan identifikasi beberapa ekspresi seperti senang, marah, takut, dan sedih. Namun, model ini menunjukkan adanya tanda-tanda overfitting, yang disebabkan oleh beberapa faktor, seperti kualitas gambar yang rendah, kemiripan antara ekspresi wajah yang berbeda, adanya bias dalam data latih yang membuat model kesulitan dalam membedakan ekspresi, atau keterbatasan model *MobileNet* itu sendiri dalam mendeteksi fitur wajah dengan akurat. Untuk mengatasi overfitting ini, salah satu solusi yang dapat diterapkan adalah menggunakan checkpoint, yang memungkinkan penyimpanan model pada titik-titik tertentu selama pelatihan untuk mencegah model terlalu menyesuaikan diri dengan data pelatihan.
2. Hasil akurasi dari Metode *CNN MobileNet* dalam deteksi ekspresi dengan epoch 150 menggambarkan distribusi prediksi model untuk setiap kelas, dengan diagonal utama menunjukkan jumlah prediksi yang benar untuk masing-masing kelas. *Precision*, *recall*, dan *f1-score* untuk setiap kelas memberikan detail lebih lanjut mengenai kinerja model. Untuk kelas Marah, *Precision* adalah 0.68, *recall* 0.73, dan *f1-score* 0.70. Pada kelas Sedih, *Precision* mencapai 0.85, *recall* 0.49, dan *f1-score* 0.62. Untuk kelas Senang, *Precision* adalah 0.94, *recall* 0.92, dan *f1-score* 0.93. Sedangkan untuk kelas Takut, *Precision* adalah 0.67, *recall* 0.89, dan *f1-score* 0.76. Model memiliki performa terbaik dalam mengklasifikasikan emosi "Senang" dengan *f1-score* 0.93, sedangkan kinerja terburuk dalam mengklasifikasikan emosi "Sedih" dengan *recall* hanya 0.49, yang menunjukkan model sering salah dalam mengidentifikasi emosi ini. Rata-rata akurasi keseluruhan model adalah 0.78, dengan *Precision* rata-rata (macro avg) 0.78, *recall* rata-rata 0.76, dan *f1-score* rata-rata 0.75.

B. SARAN

Berdasarkan hasil penelitian, penulis merekomendasikan agar penelitian berikutnya mengeksplorasi berbagai arsitektur *CNN*, seperti ResNet, Inception, atau EfficientNet, untuk menemukan model yang lebih optimal dalam mendeteksi ekspresi wajah. Selain itu, evaluasi sebaiknya dilakukan dengan menggunakan dataset yang lebih beragam, mencakup variasi dalam ekspresi, etnis, usia, dan kondisi pencahayaan, untuk memastikan kemampuan generalisasi model dalam berbagai situasi nyata. Penelitian lanjutan juga disarankan untuk memasukkan pengambilan gambar secara natural di tempat umum sebagai bagian dari dataset. Gambar yang diambil dalam lingkungan alami, seperti di jalan, pusat perbelanjaan, atau tempat kerja, akan menghadirkan variasi yang lebih realistis dalam hal sudut pandang, pencahayaan, dan ekspresi spontan, sehingga dapat meningkatkan model. Terakhir, penting untuk mempertimbangkan implementasi dan pengujian model pada data video real-time untuk menilai kinerja sistem dalam kondisi dinamis dan memastikan efisiensi operasional model dalam aplikasi dunia nyata.

DAFTAR PUSTAKA

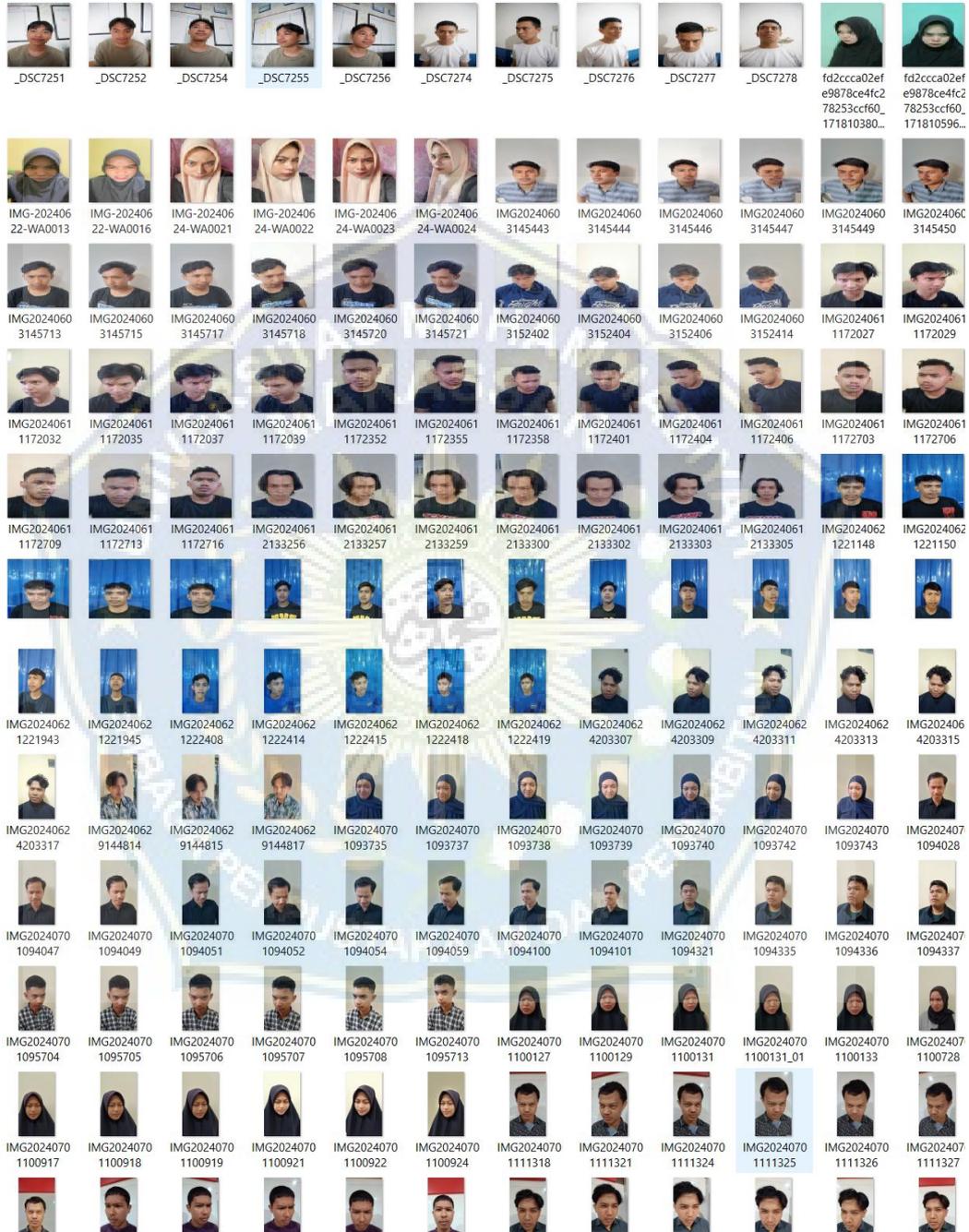
- Achmad, Y., Wihandika, R. C., & Dewi, C. (2019). Klasifikasi emosi berdasarkan ciri wajah wenggunakan convolutional neural network. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 3(11), 10595–10604.
- Daffa Ulhaq, M. R., Zaidan, M. A., & Firdaus, D. (2023). Pengenalan Ekspresi Wajah Secara Real-Time Menggunakan Metode SSD *MobileNet* Berbasis Android. *Journal of Technology and Informatics (JoTI)*, 5(1), 48–52. <https://doi.org/10.37802/joti.v5i1.387>
- DULI, S. N. H. (2024). *ANALISIS MACHINE LEARNING DAN DEEP LEARNING TERHADAP UKURAN DATASET STUDY KASUS ANALISIS SENTIMEN TEXT*.
- Fasounaki, M., Yüce, E. B., Öncül, S., & Ince, G. (2021). CNN-based Text-independent Automatic Speaker Identification Using Short Utterances. *Proceedings - 6th International Conference on Computer Science and Engineering, UBMK 2021, 01*, 413–418. <https://doi.org/10.1109/UBMK52708.2021.9559031>
- Gupta, Akanksha, Narwaria, & Pratap, D. R. (2021). Review on *Deep learning* Handwritten Digit Recognition using Convolutional Neural Network. *International Journal of Recent Technology and Engineering (IJRTE)*, 9(5), 245–247. <https://doi.org/10.35940/ijrte.e5287.019521>
- Komputasi, J. I., No, V., Ssd, M., *MobileNet*, V., & Model, S. (2020). Pembuatan Aplikasi Deteksi Objek Menggunakan *TensorFlow* Object Detection API dengan Memanfaatkan SSD *MobileNet* V2 Sebagai Model Pra - Terlatih. *Jurnal Ilmiah Komputasi*, 19(3), 421–430. <https://doi.org/10.32409/jikstik.19.3.68>
- Musa, P., Anam, W. K., Musa, S. B., Aryunani, W., Senjaya, R., & Sularsih, P. (2023). Pembelajaran Mendalam Pengklasifikasi Ekspresi Wajah Manusia dengan Model Arsitektur Xception pada Metode Convolutional Neural Network. *Rekayasa*, 16(1), 65–73. <https://doi.org/10.21107/rekayasa.v16i1.16974>

- Muttaqiin, A. K., Yuana, H., & Chulkamdi, M. T. (2023). Implementasi Algoritma *Convolutional Neural Network* Untuk Pengenalan Ekspresi Wajah. *Jurnal Riset Sistem Informasi Dan Teknik Informatika (JURASIK)*, 8(2), 772–792. <https://tunasbangsa.ac.id/ejurnal/index.php/jurasik>
- Nugroho, P. A., Fenriana, I., & Ariyanto, R. (2020). Implementasi *Deep learning* Menggunakan *Convolutional Neural Network (CNN)* Pada Ekspresi Manusia. *Algor*, 2(1), 12–21.
- Peryanto, A., Yudhana, A., & Umar, R. (2020). Klasifikasi Citra Menggunakan *Convolutional Neural Network* dan K Fold Cross Validation. *Journal of Applied Informatics and Computing*, 4(1), 45–51. <https://doi.org/10.30871/jaic.v4i1.2017>
- Rafiedhia Pramutighna, I., & Hermawan, A. (2023). Pengenalan Potensi Racun dan Peningkatan Keamanan Pangan Dalam Jamur Menggunakan *Convolutional Neural Network*. *Jurnal Media Informatika Budidarma*, 7, 1716–1726. <https://doi.org/10.30865/mib.v7i4.6372>
- Rima Dias Ramadhani, Nur Aziz Thohari, A., Kartiko, C., Junaidi, A., Ginanjar Laksana, T., & Alim Setya Nugraha, N. (2021). Optimasi Akurasi Metode *Convolutional Neural Network* untuk Identifikasi Jenis Sampah. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, 5(2), 312–318. <https://doi.org/10.29207/resti.v5i2.2754>
- Rohim, A., Sari, Y. A., & Tibyani. (2019). Convolution Neural Network (*CNN*) Untuk Pengklasifikasian Citra Makanan Tradisional. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 3(7), 7038–7042.
- Rosaly, R., & Prasetyo, A. (2019). Pengertian *TensorFlow* Beserta Fungsi dan Simbol-simbol *TensorFlow* yang Paling Umum Digunakan. <https://www.nesabamedia.com>, 2, 2. <https://www.nesabamedia.com/pengertian-TensorFlow/>
- Segal, N. (2022). *Facial Expressions Training Data*. Noam Segal. <https://www.kaggle.com/datasets/noamsegal/affectnet-training-data>
- Sena, S. (2019). *Pengenalan Deep learning Part 7 : Convolutional Neural Network*

- (CNN). <https://medium.com/@samuelsena/pengenalan-deep-learning-part-7-convolutional-neural-network-CNN-b003b477dc94>
- Syahromi. (2022). Facial Expression Recognition Menggunakan Algoritma MtCNN (Multi-Task Cascaded Convolutional Neural Network) Dan Vgg16. *Aplikasi Dan Analisis Literatur Fasilkom UI*, 4–25.
- Syamsudin. (2019). Bab II Landasan Teori. *Journal of Chemical Information and Modeling*, 53(9), 1689–1699.
- Syukron, A., Sardiarinto, S., Saputro, E., & Widodo, P. (2023). Penerapan Metode Smote Untuk Mengatasi Ketidakseimbangan Kelas Pada Prediksi Gagal Jantung. *Jurnal Teknologi Informasi Dan Terapan*, 10(1), 47–50. <https://doi.org/10.25047/jtit.v10i1.313>
- TiaraSari, A., & Haryatmi, E. (2021). Penerapan *Convolutional Neural Network Deep learning* dalam Pendeteksian Citra Biji Jagung Kering. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, 5(2), 265–271. <https://doi.org/10.29207/resti.v5i2.3040>
- Tiyar, R. I., & Fudholi, D. H. (2021). Kajian Pengaruh Dataset dan Bias Dataset terhadap Performa Akurasi Deteksi Objek. *Petir*, 14(2), 258–268. <https://doi.org/10.33322/petir.v14i2.1350>
- Wakhidah, N. (2010). Clustering Menggunakan K-Means Algorithm. *Jurnal Transformatika*, 8(1), 33. <https://doi.org/10.26623/transformatika.v8i1.45>
- Widodo, S., Setiawan, D., Ridwan, T., & Ambari, R. (2022). Perancangan Deteksi Emosi Manusia berdasarkan Ekspresi Wajah Menggunakan Algoritma VGG16. *Syntax : Jurnal Informatika*, 11(01), 01–12. <https://doi.org/10.35706/syji.v11i01.6594>

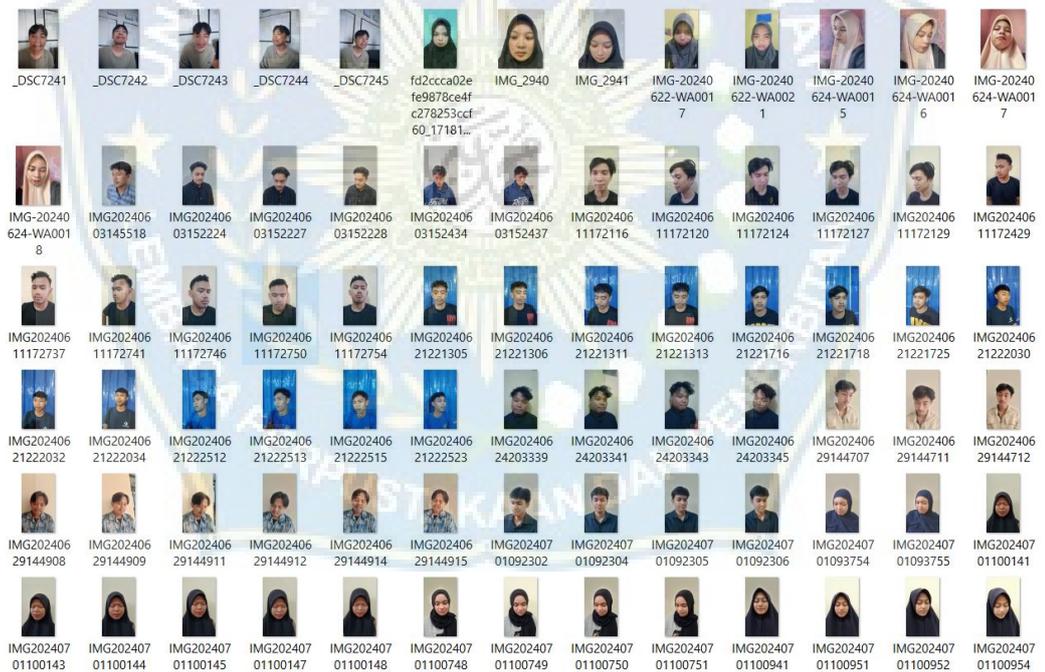
LAMPIRAN

Lampiran 1. Dataset Marah





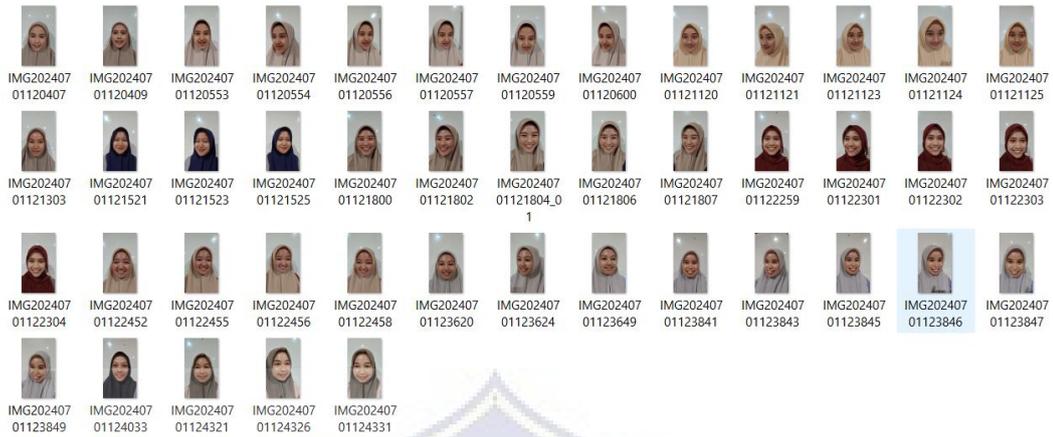
Lampiran 2. Dataset Sedih





Lampiran 3. Dataset Senang





Lampiran 4. Dataset Takut





Lampiran 5. Source Code

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

```
[ ] import os
import cv2
import xml.etree.ElementTree as ET
import numpy as np
import matplotlib.pyplot as plt
from tensorflow.keras.applications import MobileNet
from tensorflow.keras.models import Model
from tensorflow.keras.layers import Dense, GlobalAveragePooling2D
from tensorflow.keras.optimizers import Adam
from sklearn.model_selection import train_test_split
from collections import Counter
```

▶ # Path to your dataset directory
dataset_dir = '/content/drive/MyDrive/Dataset/arya6'

```
[ ] # Fungsi untuk memuat gambar dan anotasi
def load_dataset(dataset_dir):
    images = []
    annotations = []
    classes_set = set()

    # Memuat gambar dan anotasi XML yang sesuai
    image_dir = os.path.join(dataset_dir, 'images')
    label_dir = os.path.join(dataset_dir, 'labels')

    for filename in os.listdir(image_dir):
        if filename.endswith('.jpg'):
            img_path = os.path.join(image_dir, filename)
            annotation_path = os.path.join(label_dir, os.path.splitext(filename)[0] + '.xml')

            # Membaca gambar
            image = cv2.imread(img_path)
            height, width, _ = image.shape

            # parsing anotasi XML
            tree = ET.parse(annotation_path)
            root = tree.getroot()

            # Mengambil ukuran gambar dari XML
            image_size = root.find('size')
            img_width = int(image_size.find('width').text)
            img_height = int(image_size.find('height').text)

            # Mengambil koordinat bounding box dan label
            for obj in root.findall('object'):
                obj_name = obj.find('name').text
                classes_set.add(obj_name) # Menambahkan kelas ke set kelas
                bbox = obj.find('bndbox')
                xmin = int(bbox.find('xmin').text) / img_width * 320
                xmax = int(bbox.find('xmax').text) / img_width * 320
                ymin = int(bbox.find('ymin').text) / img_height * 320
                ymax = int(bbox.find('ymax').text) / img_height * 320

                annotations.append({
                    'image_path': img_path,
                    'label': obj_name,
                    'bbox': [xmin, ymin, xmax, ymax],
                })

            images.append(image)

    classes_list = sorted(list(classes_set)) # Mengubah set menjadi daftar kelas yang terurut
    num_classes = len(classes_list)
```

```
[ ] # Fungsi untuk menampilkan gambar dengan bounding box
def display_images_with_bboxes(images, annotations, classes_list, num_samples=5):
    plt.figure(figsize=(15, 15))
    for i in range(num_samples):
        image = images[i]
        ann = annotations[i]
        bbox = ann['bbox']
        label = ann['label']
        plt.subplot(1, num_samples, i + 1)
        plt.imshow(cv2.cvtColor(image, cv2.COLOR_BGR2RGB))
        plt.gca().add_patch(plt.Rectangle((bbox[0], bbox[1]), bbox[2] - bbox[0], bbox[3] - bbox[1],
                                         edgecolor='red', facecolor='none', linewidth=2))

        plt.title(classes_list[label])
    plt.show()

[ ] # Memuat dataset
images, annotations, num_classes, classes_list = load_dataset(dataset_dir)

[ ] # Mengubah label menjadi nilai numerik
class_to_index = {cls: idx for idx, cls in enumerate(classes_list)}
for annotation in annotations:
    annotation['label'] = class_to_index[annotation['label']]

[ ] # Menghitung jumlah setiap kelas dalam dataset
label_counts = Counter([ann['label'] for ann in annotations])

# Menampilkan statistik jumlah dataset berdasarkan kelas
print("\nStatistik jumlah dataset berdasarkan kelas:")
for cls, count in label_counts.items():
    print(f"{classes_list[cls]}: {count} gambar")

[ ] # Mempersiapkan data untuk pelatihan
X = np.array(images)
y_bbox = np.array([ann['bbox'] for ann in annotations])
y_class = np.array([ann['label'] for ann in annotations])

[ ] # Membagi data menjadi set pelatihan dan validasi
X_train, X_val, y_bbox_train, y_bbox_val, y_class_train, y_class_val, annotations_train, annotations_val = train_test_split(
    X, y_bbox, y_class, annotations, test_size=0.2, random_state=42)

# Menampilkan beberapa gambar dengan bounding box
display_images_with_bboxes(X_train, annotations_train, classes_list, num_samples=5)

[ ] # Menampilkan informasi dataset
print(f"Jumlah gambar: {len(X)}")
print(f"Jumlah kelas: {num_classes}")
print(f"Dimensi gambar: {X.shape[1]}")
print(f"Label kelas: {classes_list}")

[ ] # Mendefinisikan model MobileNet
base_model = MobileNet(weights='imagenet', include_top=False, input_shape=(320, 320, 3))

[ ] # Menambahkan layer custom untuk regresi bounding box dan prediksi kelas
x = base_model.output
x = GlobalAveragePooling2D()(x)
```

```

[ ] # Output untuk regresi bounding box
    bbox_output = Dense(4, activation='linear', name='bbox_output')(x)

[ ] # Output untuk prediksi kelas
    class_output = Dense(num_classes, activation='softmax', name='class_output')(x)

[ ] # Menggabungkan output
    model = Model(inputs=base_model.input, outputs=[bbox_output, class_output])

[ ] # Menampilkan bentuk model
    model.summary()

[ ] # Mengompilasi model
    model.compile(optimizer=Adam(learning_rate=0.001),
                  loss={'bbox_output': 'mean_squared_error', 'class_output': 'sparse_categorical_crossentropy'},
                  metrics={'bbox_output': 'mse', 'class_output': 'accuracy'})

[ ] #from tensorflow.keras.callbacks import EarlyStopping
    # Definisikan Early Stopping
    #early_stopping = EarlyStopping(monitor='val_loss', patience=5, restore_best_weights=True)

    from tensorflow.keras.callbacks import ModelCheckpoint

    # Path untuk menyimpan model
    checkpoint_path = '/content/drive/MyDrive/Dataset/arya6/model_checkpoint2.h5'

    # Callback untuk menyimpan model ketika val_accuracy meningkat
    checkpoint = ModelCheckpoint(
        checkpoint_path,
        monitor='val_class_output_accuracy',
        verbose=1,
        save_best_only=True,
        mode='max',
        save_weights_only=False,
        save_freq='epoch'
    )

    # Melatih model dengan ModelCheckpoint
    history = model.fit(
        X_train, {'bbox_output': y_bbox_train, 'class_output': y_class_train},
        validation_data=(X_val, {'bbox_output': y_bbox_val, 'class_output': y_class_val}),
        epochs=150,
        batch_size=32,
        callbacks=[checkpoint] # Menambahkan callback ModelCheckpoint
    )

[ ] import json

    # Simpan classes_list ke dalam file JSON di Google Drive
    def save_classes_list(classes_list, save_path):
        with open(save_path, 'w') as f:
            json.dump(classes_list, f)
            print(f"File JSON berhasil disimpan di: {save_path}")

    # Contoh penggunaan:
    # Ganti path sesuai dengan direktori yang Anda inginkan di Google Drive
    save_path = '/content/drive/MyDrive/Dataset/arya6/classes_list.json'
    save_classes_list(classes_list, save_path)

```

```
[ ] import pickle

# Daftar kelas yang digunakan selama pelatihan
classes_list = [...] # Daftar kelas yang sama dengan yang digunakan saat pelatihan

# Simpan daftar kelas ke file PKL
with open('/content/drive/MyDrive/Dataset/arya6/classes_list.pkl', 'wb') as f:
    pickle.dump(classes_list, f)
```

```
[ ] # Muat daftar kelas dari file JSON
with open('/content/drive/MyDrive/Dataset/arya6/classes_list.json', 'r') as f:
    classes_list = json.load(f)

print(classes_list)
```

```
[ ] # Menampilkan history pelatihan
plt.figure(figsize=(12, 4))
plt.subplot(1, 2, 1)
plt.plot(history.history['bbox_output_loss'], label='Bounding Box Loss')
plt.plot(history.history['val_bbox_output_loss'], label='Val Bounding Box Loss')
plt.legend()
plt.title('Bounding Box Loss')

plt.subplot(1, 2, 2)
plt.plot(history.history['class_output_accuracy'], label='Class Accuracy')
plt.plot(history.history['val_class_output_accuracy'], label='Val Class Accuracy')
plt.legend()
plt.title('Class Accuracy')
plt.show()
```

```
[ ] # Fungsi untuk menampilkan gambar dengan bounding box dan nilai prediksi
def display_validation_results(X_val, y_bbox_val, y_class_val, classes_list, model, num_samples=10, samples_per_row=5):
    predictions = model.predict(X_val)
    bbox_predictions, class_predictions = predictions[0], predictions[1]

    num_rows = (num_samples + samples_per_row - 1) // samples_per_row # Menghitung jumlah baris yang diperlukan
    plt.figure(figsize=(15, 3 * num_rows)) # Menyesuaikan ukuran gambar berdasarkan jumlah baris

    for i in range(num_samples):
        image = X_val[i]
        bbox_pred = bbox_predictions[i]
        class_pred = np.argmax(class_predictions[i])

        plt.subplot(num_rows, samples_per_row, i + 1)
        plt.imshow(cv2.cvtColor(image.astype('uint8'), cv2.COLOR_BGR2RGB))

        # Menggambar bounding box berdasarkan prediksi
        xmin, ymin, xmax, ymax = bbox_pred
        plt.gca().add_patch(plt.Rectangle((xmin, ymin), xmax - xmin, ymax - ymin,
                                         edgecolor='green', facecolor='none', linewidth=2))
        plt.title(f'{classes_list[class_pred]}\nActual: {classes_list[y_class_val[i]}'

        # Menambahkan nilai akurasi di subplot
        plt.xlabel(f'Class Accuracy: {class_predictions[i][y_class_val[i]].2f}')

    # Memastikan tidak ada subplot yang kosong di akhir baris terakhir
    if (i + 1) % samples_per_row == 0:
        plt.tight_layout()

    plt.tight_layout()
    plt.show()

# Panggil fungsi ini setelah Anda melatih model dan ingin menampilkan hasil validasi
display_validation_results(X_val, y_bbox_val, y_class_val, classes_list, model, num_samples=10, samples_per_row=5)
```

```
[ ] from sklearn.metrics import classification_report, confusion_matrix

# Prediksi kelas pada data validasi
y_pred = model.predict(X_val)

# Mendapatkan kelas prediksi dengan nilai tertinggi
y_pred_class = np.argmax(y_pred[1], axis=1)

# Menampilkan confusion matrix
cm = confusion_matrix(y_class_val, y_pred_class)
print("Confusion Matrix:")
print(cm)

# Menampilkan classification report
cr = classification_report(y_class_val, y_pred_class, target_names=classes_list)
print("Classification Report:")
print(cr)
```

```
[ ] import cv2
import numpy as np
import matplotlib.pyplot as plt

# Fungsi untuk memuat dan menampilkan gambar dengan prediksi bounding box
def validate_single_image(image_path, model, classes_list):
    # Memuat gambar dari path
    image = cv2.imread(image_path)
    image_rgb = cv2.cvtColor(image, cv2.COLOR_BGR2RGB) # Konversi ke format RGB

    # Reshape gambar untuk model
    input_image = cv2.resize(image, (320, 320)) # Ubah sesuai dengan dimensi masukan model Anda
    input_image = np.expand_dims(input_image, axis=0)

    # Melakukan prediksi menggunakan model
    predictions = model.predict(input_image)
    bbox_predictions, class_predictions = predictions[0], predictions[1]

    # Mengambil hasil prediksi untuk gambar pertama
    bbox_pred = bbox_predictions[0]
    class_pred = np.argmax(class_predictions[0])

    # Menampilkan gambar dengan bounding box dan label prediksi
    plt.figure(figsize=(8, 6))
    plt.imshow(image_rgb)

    # Menggambar bounding box berdasarkan prediksi
    xmin, ymin, xmax, ymax = bbox_pred
    plt.gca().add_patch(plt.Rectangle((xmin, ymin), xmax - xmin, ymax - ymin,
                                     edgecolor='green', facecolor='none', linewidth=2))
    plt.title(f'Predicted Class: {classes_list[class_pred]}')
    plt.axis('off')
    plt.show()

# Contoh penggunaan
# Pastikan path gambar sesuai dengan lokasi gambar yang ingin Anda uji
image_path = './content/drive/MyDrive/Dataset/arya5/images/IMG-20240622-WA0013.jpg.rf.6cb01f7f6d0b91d12075f864a134ae25.jpg' # Ganti dengan path gambar Anda
validate_single_image(image_path, model, classes_list)
```

```

[ ] import cv2
import numpy as np
import matplotlib.pyplot as plt
from tensorflow.keras.models import load_model
import json

# Muat model yang telah dilatih
model_path = '/content/drive/MyDrive/Dataset/arya6/model_checkpoint2.h5'
model = load_model(model_path)

# Muat daftar kelas dari file JSON
with open('/content/drive/MyDrive/Dataset/arya6/classes_list.json', 'r') as f:
    classes_list = json.load(f)

# Fungsi untuk memuat gambar uji
def load_test_image(image_path):
    image = cv2.imread(image_path)
    image = cv2.resize(image, (320, 320))
    return image

# Fungsi untuk pra-pemrosesan gambar
def preprocess_image(image):
    #image = image / 255.0
    image = np.expand_dims(image, axis=0)
    return image

# Fungsi untuk membuat prediksi
def make_predictions(model, image):
    bbox_pred, class_pred = model.predict(image)
    class_pred = np.argmax(class_pred, axis=-1)
    return bbox_pred, class_pred

# Fungsi untuk menampilkan hasil
def display_test_image(image_path, bbox_pred, class_pred, classes_list):
    # Debugging: Periksa prediksi kelas dan daftar kelas
    print(f"class_pred: {class_pred}")
    print(f"classes_list: {classes_list}")

    if class_pred[0] < len(classes_list):
        label = classes_list[class_pred[0]]
    else:
        label = "Unknown"

    image = cv2.imread(image_path)
    image = cv2.resize(image, (320, 320))
    bbox = bbox_pred[0]

```

```

plt.figure(figsize=(10, 10))
plt.imshow(cv2.cvtColor(image, cv2.COLOR_BGR2RGB))
plt.gca().add_patch(plt.Rectangle((bbox[0], bbox[1]), bbox[2] - bbox[0], bbox[3] - bbox[1],
                                edgecolor='red', facecolor='none', linewidth=2))

plt.title(label)
plt.show()

# Path ke gambar uji
test_image_path = '/content/drive/MyDrive/Dataset/Validasi 2/IMG20240712150137.jpg'

# Muat dan pra-pemroses gambar uji
test_image = load_test_image(test_image_path)
preprocessed_image = preprocess_image(test_image)

# Buat prediksi
bbox_pred, class_pred = make_predictions(model, preprocessed_image)

# Tampilkan hasil
display_test_image(test_image_path, bbox_pred, class_pred, classes_list)

```

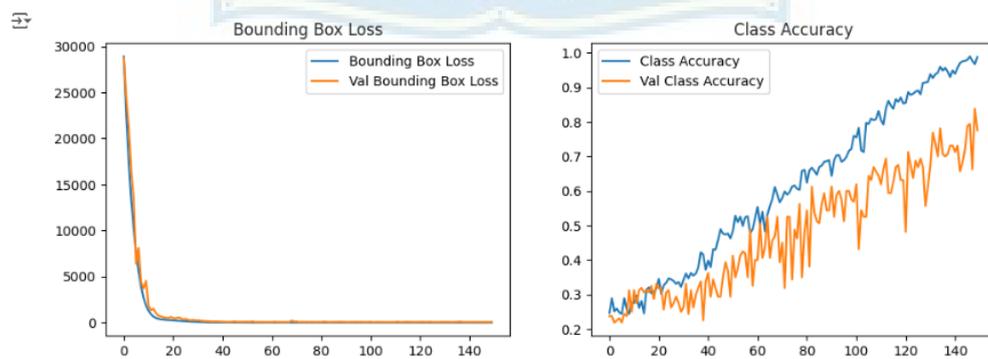
Lampiran 6. Proses Training

```

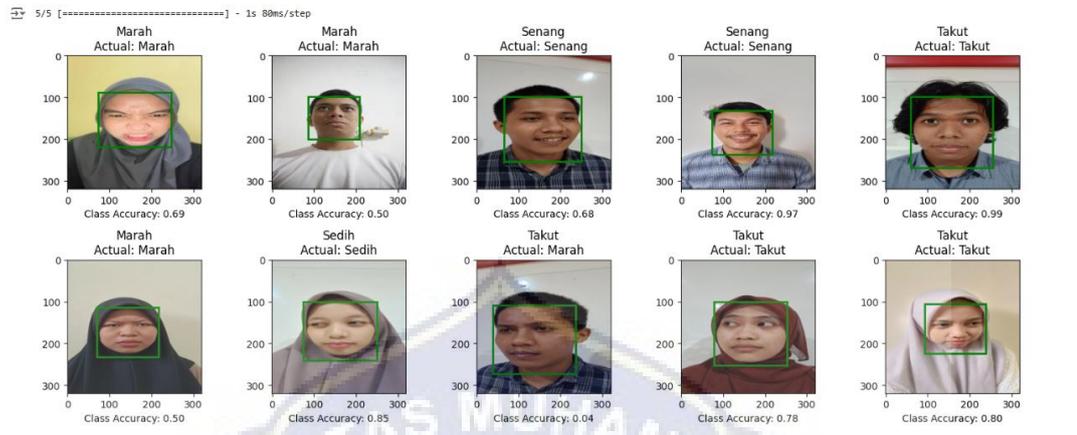
Epoch 139/150
28200 [.....] ETA: 0s - loss: 7.0233 - bbox_output_loss: 7.6653 - class_output_loss: 0.2471 - bbox_output_mse: 7.6653 - class_output_accuracy: 0.9463
Epoch 139 | val_class_output_accuracy did not improve from 0.78125
Epoch 139/150
28200 [.....] ETA: 0s - loss: 7.0233 - bbox_output_loss: 7.6653 - class_output_loss: 0.2471 - bbox_output_mse: 7.6653 - class_output_accuracy: 0.9463 - val_loss: 81.4446 - val_bbox_output_loss: 69.9599 - val_class_output_loss: 0.9941 - val_bbox
Epoch 139/150
28200 [.....] ETA: 0s - loss: 7.0233 - bbox_output_loss: 7.6653 - class_output_loss: 0.2471 - bbox_output_mse: 7.6653 - class_output_accuracy: 0.9463 - val_loss: 81.4446 - val_bbox_output_loss: 69.9599 - val_class_output_loss: 0.9941 - val_bbox
Epoch 140/150
28200 [.....] ETA: 0s - loss: 6.1132 - bbox_output_loss: 5.8827 - class_output_loss: 0.2385 - bbox_output_mse: 5.8827 - class_output_accuracy: 0.9484
Epoch 140 | val_class_output_accuracy did not improve from 0.78125
Epoch 140/150
28200 [.....] ETA: 0s - loss: 6.1132 - bbox_output_loss: 5.8827 - class_output_loss: 0.2385 - bbox_output_mse: 5.8827 - class_output_accuracy: 0.9484 - val_loss: 69.1459 - val_bbox_output_loss: 62.4725 - val_class_output_loss: 0.6764 - val_bbox
Epoch 141/150
28200 [.....] ETA: 0s - loss: 5.3865 - bbox_output_loss: 5.1599 - class_output_loss: 0.2265 - bbox_output_mse: 5.1599 - class_output_accuracy: 0.9391
Epoch 141 | val_class_output_accuracy did not improve from 0.78125
Epoch 141/150
28200 [.....] ETA: 0s - loss: 5.3865 - bbox_output_loss: 5.1599 - class_output_loss: 0.2265 - bbox_output_mse: 5.1599 - class_output_accuracy: 0.9391 - val_loss: 66.8574 - val_bbox_output_loss: 65.3150 - val_class_output_loss: 0.7425 - val_bbox
Epoch 142/150
28200 [.....] ETA: 0s - loss: 6.1428 - bbox_output_loss: 5.9489 - class_output_loss: 0.2818 - bbox_output_mse: 5.9489 - class_output_accuracy: 0.9563
Epoch 142 | val_class_output_accuracy did not improve from 0.78125
Epoch 142/150
28200 [.....] ETA: 0s - loss: 6.1428 - bbox_output_loss: 5.9489 - class_output_loss: 0.2818 - bbox_output_mse: 5.9489 - class_output_accuracy: 0.9563 - val_loss: 66.2481 - val_bbox_output_loss: 65.6327 - val_class_output_loss: 0.6884 - val_bbox
Epoch 143/150
28200 [.....] ETA: 0s - loss: 6.1725 - bbox_output_loss: 5.9797 - class_output_loss: 0.2826 - bbox_output_mse: 5.9797 - class_output_accuracy: 0.9703
Epoch 143 | val_class_output_accuracy did not improve from 0.78125
Epoch 143/150
28200 [.....] ETA: 0s - loss: 6.1725 - bbox_output_loss: 5.9797 - class_output_loss: 0.2826 - bbox_output_mse: 5.9797 - class_output_accuracy: 0.9703 - val_loss: 77.1646 - val_bbox_output_loss: 76.1554 - val_class_output_loss: 1.0113 - val_bbox
Epoch 144/150
28200 [.....] ETA: 0s - loss: 6.5491 - bbox_output_loss: 6.3623 - class_output_loss: 0.1849 - bbox_output_mse: 6.3623 - class_output_accuracy: 0.9750
Epoch 144 | val_class_output_accuracy did not improve from 0.78125
Epoch 144/150
28200 [.....] ETA: 0s - loss: 6.5491 - bbox_output_loss: 6.3623 - class_output_loss: 0.1849 - bbox_output_mse: 6.3623 - class_output_accuracy: 0.9750 - val_loss: 62.6375 - val_bbox_output_loss: 61.9356 - val_class_output_loss: 0.7891 - val_bbox
Epoch 145/150
28200 [.....] ETA: 0s - loss: 5.4585 - bbox_output_loss: 5.2791 - class_output_loss: 0.1784 - bbox_output_mse: 5.2791 - class_output_accuracy: 0.9766
Epoch 145 | val_class_output_accuracy did not improve from 0.78125
Epoch 145/150
28200 [.....] ETA: 0s - loss: 5.4585 - bbox_output_loss: 5.2791 - class_output_loss: 0.1784 - bbox_output_mse: 5.2791 - class_output_accuracy: 0.9766 - val_loss: 72.8553 - val_bbox_output_loss: 72.1756 - val_class_output_loss: 0.6767 - val_bbox
Epoch 146/150
28200 [.....] ETA: 0s - loss: 4.4630 - bbox_output_loss: 3.9089 - class_output_loss: 0.5516 - bbox_output_mse: 3.9089 - class_output_accuracy: 0.9797
Epoch 146 | val_class_output_accuracy improved from 0.78125 to 0.78750, saving model to /content/drive/MyDrive/Dataset/valid_model_checkpoint1215
Epoch 146/150
28200 [.....] ETA: 0s - loss: 4.4630 - bbox_output_loss: 3.9089 - class_output_loss: 0.5516 - bbox_output_mse: 3.9089 - class_output_accuracy: 0.9797 - val_loss: 66.2611 - val_bbox_output_loss: 65.6628 - val_class_output_loss: 0.5793 - val_bbox
Epoch 147/150
28200 [.....] ETA: 0s - loss: 4.7354 - bbox_output_loss: 4.4822 - class_output_loss: 0.1311 - bbox_output_mse: 4.4822 - class_output_accuracy: 0.9891
Epoch 147 | val_class_output_accuracy improved from 0.78750 to 0.79375, saving model to /content/drive/MyDrive/Dataset/valid_model_checkpoint1215
Epoch 147/150
28200 [.....] ETA: 0s - loss: 4.7354 - bbox_output_loss: 4.4822 - class_output_loss: 0.1311 - bbox_output_mse: 4.4822 - class_output_accuracy: 0.9891 - val_loss: 69.6272 - val_bbox_output_loss: 68.5458 - val_class_output_loss: 0.5283 - val_bbox
Epoch 148/150
28200 [.....] ETA: 0s - loss: 3.8147 - bbox_output_loss: 3.7683 - class_output_loss: 0.1464 - bbox_output_mse: 3.7683 - class_output_accuracy: 0.9766
Epoch 148 | val_class_output_accuracy did not improve from 0.79375
Epoch 148/150
28200 [.....] ETA: 0s - loss: 3.8147 - bbox_output_loss: 3.7683 - class_output_loss: 0.1464 - bbox_output_mse: 3.7683 - class_output_accuracy: 0.9766 - val_loss: 72.9437 - val_bbox_output_loss: 71.2167 - val_class_output_loss: 0.8258 - val_bbox
Epoch 149/150
28200 [.....] ETA: 0s - loss: 3.7864 - bbox_output_loss: 3.6321 - class_output_loss: 0.2333 - bbox_output_mse: 3.6321 - class_output_accuracy: 0.9672
Epoch 149 | val_class_output_accuracy improved from 0.79375 to 0.83750, saving model to /content/drive/MyDrive/Dataset/valid_model_checkpoint1215
Epoch 149/150
28200 [.....] ETA: 0s - loss: 3.7864 - bbox_output_loss: 3.6321 - class_output_loss: 0.2333 - bbox_output_mse: 3.6321 - class_output_accuracy: 0.9672 - val_loss: 67.4245 - val_bbox_output_loss: 67.8999 - val_class_output_loss: 0.5246 - val_bbox
Epoch 150/150
28200 [.....] ETA: 0s - loss: 5.4889 - bbox_output_loss: 5.2848 - class_output_loss: 0.1249 - bbox_output_mse: 5.2848 - class_output_accuracy: 0.9875
Epoch 150 | val_class_output_accuracy did not improve from 0.83750
Epoch 150/150
28200 [.....] ETA: 0s - loss: 5.4889 - bbox_output_loss: 5.2848 - class_output_loss: 0.1249 - bbox_output_mse: 5.2848 - class_output_accuracy: 0.9875 - val_loss: 81.0901 - val_bbox_output_loss: 81.5691 - val_class_output_loss: 0.6218 - val_bbox

```

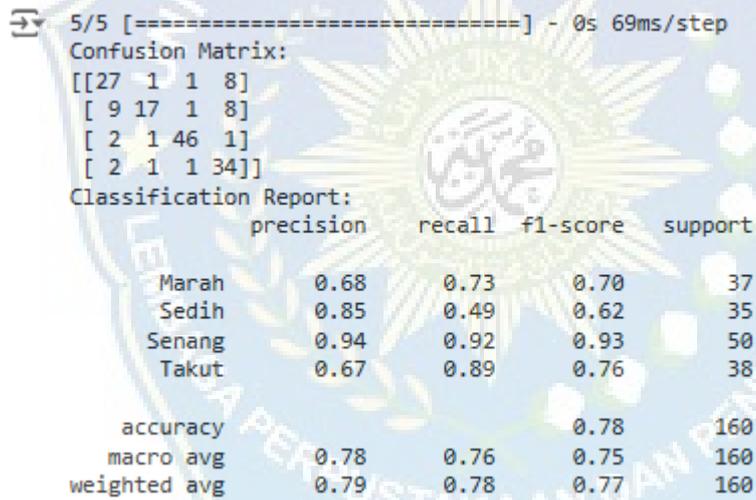
Lampiran 7. Grafik Bounding Box Loss dan Class Accuracy



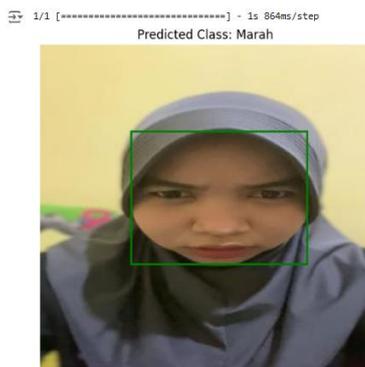
Lampiran 8. Gambar dengan bounding box dan nilai prediksi



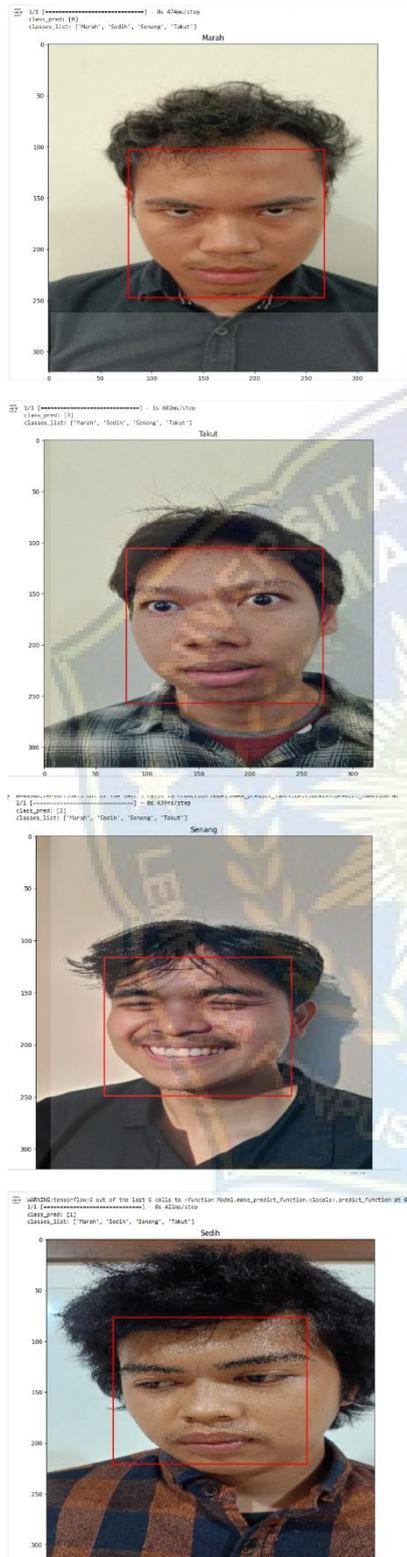
Lampiran 9. Hasil Dari Training



Lampiran 10. Gambar dengan prediksi bounding box



Lampiran 11. Hasil Deteksi Gambar





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

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

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

SURAT KETERANGAN BEBAS PLAGIAT

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

Nama : Arya Wibawa. Ar

Nim : 105841109320

Program Studi : Teknik Informatika

Dengan nilai:

No	Bab	Nilai	Ambang Batas
1	Bab 1	10 %	10 %
2	Bab 2	21 %	25 %
3	Bab 3	10 %	10 %
4	Bab 4	6 %	10 %
5	Bab 5	5 %	5 %

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

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

Makassar, 13 Agustus 2024

Mengetahui,

Kepala UPT- Perpustakaan dan Penerbitan,



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

BAB I Arya Wibawa. Ar - 105841109320

by Tahap Tutup



Submission date: 12-Aug-2024 01:20PM (UTC+0700)

Submission ID: 2430875445

File name: BAB_1_-_2024-08-12T141938.563.docx (22.32K)

Word count: 740

Character count: 4975

BAB I Arya Wibawa. Ar - 105841109320

ORIGINALITY REPORT

10%

SIMILARITY INDEX

10%

INTERNET SOURCES

2%

PUBLICATIONS

%

STUDENT PAPERS

PRIMARY SOURCES

1

repository.binadarma.ac.id

Internet Source

3%

2

123dok.com

Internet Source

2%

3

core.ac.uk

Internet Source

2%

4

blogsainulh.wordpress.com

Internet Source

2%

5

docplayer.info

Internet Source

2%

Exclude quotes On

Exclude matches < 2%

Exclude bibliography On

BAB II Arya Wibawa. Ar -
105841109320
by Tahap Tutup



Submission date: 12-Aug-2024 01:22PM (UTC+0700)

Submission ID: 2430876297

File name: BAB_II_-_2024-08-12T141939.061.docx (571.93K)

Word count: 2236

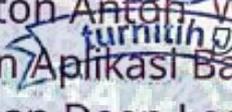
Character count: 15004

BAB II Arya Wibawa. Ar - 105841109320

ORIGINALITY REPORT

21 %	19 %	2 %	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	eprints.itenas.ac.id Internet Source		17 %
2	Robby Kamil Siregar, Anton Anton Widiastuti Widiastuti. "Perancangan Aplikasi Bahasa Isyarat "Isyaratku" Dengan Deep Learning Serta Google Cloud Platform", Simpatik: Jurnal Sistem Informasi dan Informatika, 2021 Publication		2 %
3	doaj.org Internet Source		2 %

Exclude quotes On Exclude matches < 2%
Exclude bibliography On

BAB III Arya Wibawa. Ar - 105841109320

by Tahap Tutup



Submission date: 09-Aug-2024 03:34PM (UTC+0700)

Submission ID: 2429455028

File name: BAB_III_-_2024-08-09T163131.056.docx (40.36K)

Word count: 1404

Character count: 9315

BAB III Arya Wibawa. Ar - 105841109320

ORIGINALITY REPORT

10% SIMILARITY INDEX	10% INTERNET SOURCES	5% PUBLICATIONS	% STUDENT PAPERS
--------------------------------	--------------------------------	---------------------------	----------------------------

PRIMARY SOURCES

1	digilibadmin.unismuh.ac.id Internet Source	5%
2	repository.ar-raniry.ac.id Internet Source	3%
3	eprints.umsb.ac.id Internet Source	2%

Exclude quotes On
Exclude bibliography On

Exclude matches < 2%

BAB IV Arya Wibawa. Ar - 105841109320

by Tahap Tutup



Submission date: 09-Aug-2024 03:36PM (UTC+0700)

Submission ID: 2429455524

File name: BAB_VI_33.docx (5.82M)

Word count: 4864

Character count: 35862

ORIGINALITY REPORT

6%

SIMILARITY INDEX

6%

INTERNET SOURCES

6%

PUBLICATIONS

%

STUDENT PAPERS

PRIMARY SOURCES

1	digilibadmin.unismuh.ac.id Internet Source	2%
2	check-this.tistory.com Internet Source	1%
3	programtalk.com Internet Source	1%
4	www.thiscodeworks.com Internet Source	1%
5	Ryan Ali Mas'ud, Junta Zeniarja. "Optimasi Convolutional Neural Networks untuk Deteksi Kanker Payudara menggunakan Arsitektur DenseNet", Edumatic: Jurnal Pendidikan Informatika, 2024 Publication	1%
6	csafe-isu.github.io Internet Source	1%
7	shell-mag.com Internet Source	1%
8	github.com Internet Source	

BAB V Arya Wibawa. Ar - 105841109320

by Tahap Tutup



Submission date: 12-Aug-2024 01:53PM (UTC+0700)

Submission ID: 2430893173

File name: BAB_V_-_2024-08-12T141944.050.docx (15.37K)

Word count: 284

Character count: 1894

BAB V Arya Wibawa. Ar - 105841109320

ORIGINALITY REPORT

5%

SIMILARITY INDEX

5%

INTERNET SOURCES

0%

PUBLICATIONS

%

STUDENT PAPERS

PRIMARY SOURCES

1

romanpub.com

Internet Source

3%

2

repository.ub.ac.id

Internet Source

3%



Exclude quotes On

Exclude matches < 2%

Exclude bibliography On

