

**PENGUKURAN TINGKAT AKURASI DETEKSI WAJAH
AUTIS MENGGUNAKAN CNN (*CONVOLUTIONAL
NEURAL NETWORK*)**

TUGAS AKHIR

Diajukan Oleh:

MAI JEFRI

190705095

**Mahasiswa Fakultas Sains dan Teknologi
Program Studi Teknologi Informasi**



**FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI AR-RANIRY
BANDA ACEH
2024 M / 1446 H**

LEMBAR PERSETUJUAN

**PENGUKURAN TINGKAT AKURASI DETEKSI WAJAH
AUTIS MENGGUNAKAN CNN (*CONVOLUTIONAL
NEURAL NETWORK*)**

TUGAS AKHIR

Diajukan kepada Fakultas Sains dan Teknologi UIN Ar-Raniry Banda Aceh
Sebagai Salah Satu Persyaratan Penulisan Tugas Akhir/Skripsi
dalam Ilmu/Prodi Teknologi Informasi

Oleh:

**MAI JEFRI
190705095**

**Mahasiswa Fakultas Sains dan Teknologi
Program Studi Teknologi Informasi**

Disetujui Untuk Dimunaqasyahkan Oleh:

Pembimbing I,



Malahayati. M.T

NIP. 198301272015032003

Pembimbing II,



Khairan/AR. M.Kom

NIP. 198607042014031001

Mengetahui

Ketua Program Studi Teknologi Informasi,



Malahayati. M.T

NIP. 198301272015032003

LEMBAR PERSETUJUAN

PENGUKURAN TINGKAT AKURASI DETEKSI WAJAH AUTIS MENGGUNAKAN CNN (*CONVOLUTIONAL NEURAL NETWORK*)

TUGAS AKHIR

Telah Diuji Oleh Panitia Ujian Munaqasyah Tugas Akhir
Fakultas Sains an Teknologi UIN Ar-Raniry Banda Aceh dan Dinyatakan Lulus
Serta Diterima Sebagai Salah Satu Beban Studi Program Sarjana (S-1)
Dalam Prodi Teknologi Informasi

Pada Hari/Tanggal : 14 Januari 2025
4 Rajab 1446 H

Di Darussalam, Banda Aceh
Panitia Ujian Munaqasyah Tugas Akhir

Ketua,



Khairan AR. M.Kom
NIP. 198607042014031001

Sekretaris,



Malahayati. M.T
NIP. 198301272015032003

Penguji 1,



Hendri Ahmadian, M.I.M
NIP. 198301042014031002

Penguji 2,



Fathiah, M.Eng
NIP. 198606152019032010

Mengetahui,

Dekan Fakultas Sains dan Teknologi
UIN Ar-Raniry Banda Aceh



Prof. Dr. Ir. M. Dirhamsyah, M.T., IPU
NIP.196210021988111001

LEMBAR PERNYATAAN KEASLIAN SKRIPSI

Yang bertanda tangan di bawah ini :

Nama : Mai Jefri
Nim : 190705095
Program Studi : Teknologi Informasi
Fakultas : Sains dan Teknologi
Judul : Pengukuran Tingkat Akurasi Deteksi Wajah Autis
Menggunakan CNN (*Convolutional Neural Network*)

Dengan ini menyatakan bahwa dalam penulisan skripsi ini, saya:

1. Tidak menggunakan ide orang lain tanpa mampu mengembangkan dan mempertanggung jawabkan;
2. Tidak melakukan plagiasi terhadap naskah karya orang lain;
3. Tidak menggunakan karya orang lain tanpa menyebutkan sumber asli atau tanpa izin pemilik karya;
4. Tidak memanipulasi dan memalsukan data;
5. Mengerjakan sendiri karya ini dan mampu bertanggungjawab atas karya ini.

Bila dikemudian hari ada tuntutan dari pihak lain atas karya saya, dan telah melalui pembuktian yang dapat dipertanggungjawabkan dan ternyata memang ditemukan bukti bahwa saya telah melanggar pernyataan ini, maka saya siap dikenai sanksi berdasarkan aturan yang berlaku di Fakultas Sains dan Teknologi UIN Ar-Raniry Banda Aceh.

Demikian pernyataan ini saya buat dengan sesungguhnya dan tanpa paksaan dari pihak manapun.

Banda Aceh, 15 Januari 2025
Menyatakan

Mai Jefri

ABSTRAK

Nama : Mai Jefri
NIM :
Fakultas/Prodi : Sains Dan Teknologi/Teknologi Informasi
Judul : Pengukuran Tingkat Akurasi Deteksi Wajah Autis Menggunakan CNN (*Convolutional Neural Network*)

Jumlah Halaman : 69
Pembimbing I : Malahayati. M.T
Pembimbing II : Khairan AR. M.Kom

Deteksi wajah autisme adalah pendekatan teknologi yang memanfaatkan analisis wajah untuk membantu dalam diagnosis gangguan spektrum autisme. Dalam konteks deteksi wajah pada autisme, teknologi berfokus pada bagaimana individu mengekspresikan emosi melalui wajah mereka. Awalnya teknologi ini hanya mengandalkan metode berbasis fitur yang menggunakan algoritma untuk mendeteksi elemen wajah seperti mata, hidung dan mulut. Metode ini memerlukan ekstraksi fitur manual dan sering mengalami kesulitan dalam menghadapi variasi dan *noise* dalam gambar. Dari permasalahan tersebut, maka dibutuhkannya sebuah pengukur tingkat akurasi deteksi wajah anak autis. Pengukuran tingkat akurasi deteksi wajah anak autis ini menggunakan metode *Convolutional Neural Network* dengan bahasa pemrograman *python*. Dengan adanya penelitian ini hasil tingkat akurasi deteksi wajah anak autis dapat diketahui. Data yang di ambil berjumlah 2540 data anak autis 1270 dan tidak autis 1270 yang akan di uji dengan data *testing* 25% dari jumlah *dataset* yang ditemukan. Hasil yang sudah didapatkan dari penelitian ini dengan data *testing* sebanyak 300 data gambar mendapatkan nilai dari *accuracy* , *precision* , *recall* , *f1-score* dan dapat disimpulkan akurasi yang diperoleh sebesar 71%.

Kata Kunci : Teknologi, *Python*, *Autism*, *Non-Autism*, *Convolutional Neural Network*

KATA PENGANTAR

Alhamdulillah, Puji dan Syukur kita panjatkan kehadirat Allah Subhanahu Wata'ala. Dzat yang hanya kepada-Nya memohon pertolongan. Alhamdulillah atas segala pertolongan, rahmat, dan kasih sayang-Nya sehingga penulis dapat menyelesaikan tugas akhir yang berjudul **Pengukuran Tingkat Akurasi Deteksi Wajah Autis Menggunakan CNN (Convolutional Neural Network)**. Shalawat dan salam senantiasa kita kirimkan kepada Rasulullah Shallallahu Alaihi Wasallam yang senantiasa menjadi sumber inspirasi dan teladan terbaik untuk umat manusia.

Tugas akhir ini dibuat untuk memenuhi tugas akhir perkuliahan dan sebagai salah satu persyaratan untuk memperoleh gelar Sarjana Strata 1 di Program Studi Teknologi Informasi Fakultas Sains dan Teknologi Universitas Islam Negeri Ar-Raniry. Selain itu, skripsi ini juga dibuat sebagai salah satu wujud implementasi dari ilmu yang didapatkan selama masa perkuliahan di Program Studi Teknologi Informasi.

Penulis menyadari bahwa tugas akhir masih jauh dari kata sempurna. Oleh karena itu, penulis berharap dapat belajar lebih banyak lagi dalam mengimplementasikan ilmu yang didapatkan. Tugas akhir ini tentunya tidak lepas dari bimbingan, masukan, dan arahan dari berbagai pihak. Oleh karena itu, sudah sepantasnya penulis dengan penuh hormat mengucapkan terima kasih dan mendoakan semoga Allah memberikan balasan terbaik kepada:

1. Ibunda dan Ayahanda serta keluarga tercinta yang telah mendoakan, memberikan dukungan dan memotivasi dalam menyelesaikan tugas akhir ini.
2. Bapak Dr. Ir. Muhammad Dirhamsyah, M.T., IPU. Selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Ar-Raniry Banda Aceh.
3. Ibu Malahayati, M.T selaku Ketua Prodi Teknologi Informasi Fakultas Sains dan Teknolgi Universitas Islam Negeri Ar-Raniry Banda Aceh.
4. Bapak Khairan AR, M.Kom sebagai Penasehat Akademik dari semester 1 sampai semester 11.

5. Bapak Khairan AR, M.Kom selaku pembimbing 1 yang telah meluangkan waktunya untuk membimbing saya dalam menyelesaikan tugas akhir.
6. Ibu Cut Ida Rahmadiana, S.Si selaku Staff Prodi Teknologi Informasi yang senantiasa membantu penulis dalam pemberkasan administrasi.
7. Ucapan terima kasih juga kepada kawan-kawan angkatan 2019 khususnya kepada grup bagi-bagi loker yang telah banyak membantu memberikan masukan dan motivasi.
8. Terima kasih juga penulis haturkan untuk semua pihak yang telah membantu penulis dalam menyelesaikan tugas akhir ini yang tidak dapat penulis sebutkan satu persatu.

Akhir kata penulis menyadari bahwa tidak ada yang sempurna, penulis masih melakukan kesalahan dalam penyusunan tugas akhir. Oleh karena itu, penulis meminta maaf yang sedalam-dalamnya atas kesalahan yang dilakukan penulis. Penulis berharap semoga tugas akhir ini dapat bermanfaat bagi pembaca dan dapat dijadikan referensi demi pengembangan ke arah yang lebih baik. Kebenaran datangnya dari Allah dan kesalahan datangnya dari diri penulis. Semoga Allah SWT senantiasa melimpahkan Rahmat dan Ridho-Nya kepada kita semua.

Banda Aceh, 07 Januari 2024

Penulis

جامعة الرانيري

A R - R A N I R Y

Mai Jefri

DAFTAR ISI

HALAMAN JUDUL	
LEMBAR PERSETUJUAN	
LEMBAR PENGESAHAN SIDANG	
LEMBAR PERNYATAAN KEASLIAN	
ABSTRAK	iv
KATA PENGANTAR.....	ii
DAFTAR ISI.....	iv
DAFTAR GAMBAR.....	vi
DAFTAR TABEL	viii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang Masalah.....	1
1.2 Rumusan Masalah	4
1.3 Tujuan Penelitian	5
1.4 Batasan Penelitian	5
1.5 Manfaat Penelitian	5
BAB II TINJAUAN PUSTAKA.....	6
2.1 Penelitian Terdahulu	6
2.2 Kajian Teoritis.....	10
2.2.1 Pengertian Autisme.....	10
2.2.2 Deteksi Wajah dan Teknik pengolahan Citra	10
2.2.3 <i>Deep Learning</i>	11
2.2.4 VGG.....	11
2.2.5 CNN.....	12
2.2.6 <i>Confusion Matrix</i>	16
BAB III METODE PENELITIAN.....	19
3.1 Tahapan Penelitian	19
3.2 Pengumpulan Data	19
3.2.1 Jenis dan Sumber Data.....	20
3.2.2 Kebutuhan Perangkat Keras (<i>Hardware</i>)	21
3.2.3 Kebutuhan Perangkat Lunak (<i>Software</i>).....	21
3.3 Preprocessing Data.....	23
3.2.1 Penghapusan Gambar yang Tidak Jelas atau Buram	25
3.3.2 Mengubah Ukuran Gambar	25
3.3.3 Normalisasi Gambar	25
3.3.4 <i>Split Data</i>	26
3.3.5 Peningkatan Kualitas Gambar	26
3.4 Pembuatan Model <i>Deep Learning</i>	26
3.4.1 Instalasi <i>Environment</i>	28
3.4.2 <i>Input</i> Gambar	28
3.4.3 <i>Convolutional Layers</i>	28
3.4.4 <i>Pooling Layers</i>	29

3.4.5	<i>Fully Connected Layers</i>	29
3.4.6	<i>Proses Backpropagation</i>	30
3.4.7	Evaluasi Akurasi Model	31
3.5	Pelatihan dan Pengujian Model	31
3.5.1	Pelatihan Algoritma CNN	32
3.5.2	Proses Pengujian Model	32
3.5.3	Pengolahan Data Uji	33
3.5.4	Proses <i>Epoch</i>	33
BAB IV	HASIL DAN PEMBAHASAN	35
4.1	Hasil Penelitian	35
4.1.1	Persiapan Data	35
4.1.2	<i>Autism</i>	37
4.1.3	<i>Non Autism</i>	37
4.1.5	<i>Data Training</i>	38
4.1.6	<i>Data Testing</i>	38
4.1.7	Model Vgg16	38
4.1.8	Generator Pengujian Gambar	40
4.1.9	<i>Training Model</i>	40
4.1.10	Label Prediksi Gambar	42
4.1.11	Model Akurasi	42
4.1.12	Laporan Klasifikasi Pada Kinerja Model	45
4.1.13	<i>Confusion Matrix</i>	47
4.1.14	Kurfa AOC – ROC	48
4.1.15	Hasil Prediksi	49
4.1.16	Label Prediksi Gambar	51
4.1.17	Hasil Akurasi	51
4.2	Pembahasan Hasil Penelitian	53
BAB V	KESIMPULAN DAN SARAN	55
5.1	Kesimpulan	55
5.2	Saran	55
DAFTAR PUSTAKA	56

DAFTAR GAMBAR

Gambar 2.1 Arsitektur CNN	12
Gambar 2.2 <i>Convolutional Layer</i>	13
Gambar 2.3 ReLU	14
Gambar 2.4 <i>Pooling Layer</i>	15
Gambar 2.5 <i>Fully Connected Layer</i>	16
Gambar 3.1 Tahapan Penelitian	19
Gambar 3.2 Anak <i>Autism</i>	20
Gambar 3.3 Anak <i>Non Autism</i>	20
Gambar 3.4 <i>Flowchart Preprocessing Data</i>	24
Gambar 3.5 Arsitektur VVG-16 (Ashushekar, 2024)	27
Gambar 4.1 <i>Import Library</i>	35
Gambar 4.2 <i>Code Membaca Data</i>	36
Gambar 4.3 Contoh Sample Data.....	36
Gambar 4.4 Contoh Gambar <i>Autism</i>	37
Gambar 4.5 Contoh Gambar <i>Non Autism</i>	37
Gambar 4.6 Grafik <i>Dataset</i>	38
Gambar 4.7 Memanggil Model Vgg16	38
Gambar 4.8 Menyambungkan Model Vgg16	39
Gambar 4.9 Hasil Model Vgg16	39
Gambar 4.10 Generator Pengujian Gambar	40
Gambar 4.11 <i>Training Model 10 Epoch</i>	40
Gambar 4.12 <i>Training Model 20 Epoch</i>	41
Gambar 4.13 <i>Training Model 30 Epoch</i>	41
Gambar 4.14 <i>Training Model 40 Epoch</i>	41
Gambar 4.15 Label Prediksi Gambar	42
Gambar 4.16 Hasil Gambar Uji Dan Prediksi.....	42
Gambar 4.17 Model Akurasi.....	43
Gambar 4.18 <i>Training and Validation Accuracy</i>	43
Gambar 4.19 Grafik <i>Training and Validation Accuracy</i>	43
Gambar 4.20 <i>Training and Validation Loss</i>	44
Gambar 4.21 Grafik <i>Training and Validation Loss</i>	44
Gambar 4.22 Klasifikasi Kinerja Model <i>Epoch 10</i>	45
Gambar 4.23 Klasifikasi Kinerja Model <i>Epoch 20</i>	45
Gambar 4.24 Klasifikasi Kinerja Model <i>Epoch 30</i>	46
Gambar 4.25 Klasifikasi Kinerja Model <i>Epoch 40</i>	46
Gambar 4.26 Memanggil <i>Confusion Matrix</i>	47
Gambar 4.27 Visualisasi <i>Confusion Matrix</i>	47
Gambar 4.28 Hasil <i>Confusion Matrix</i>	47
Gambar 4.29 Memanggil Kurva AUC - ROC	48
Gambar 4.30 Grafik Kurva AUC - ROC	48

Gambar 4.31 Cek Hasil Prediksi.....	49
Gambar 4.32 Hasil Prediksi <i>Non-Austism</i>	50
Gambar 4.33 Hasil Prediksi <i>Austism</i>	50



DAFTAR TABEL

Tabel 2.1 Penelitian Terdahulu	8
Tabel 2.2 <i>Confusion Matrix</i>	16
Tabel 4.1 Label Prediksi Gambar.....	51
Tabel 4.2 <i>Confusion Matrix</i>	51
Tabel 4.3 Nilai Akurasi	53
Tabel 4.4 Tabel Kesimpulan	53



BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Teknologi dalam beberapa tahun terakhir telah berkembang dengan sangat cepat, mengubah hampir setiap aspek kehidupan sehari-hari manusia. Teknologi kini tidak hanya mempengaruhi cara kita berinteraksi, bekerja, dan berkomunikasi, tetapi juga membantu batasan dalam sector - sektor seperti kesehatan, pendidikan, dan industri. Inovasi dalam teknologi informasi telah membuka peluang baru, memberikan dampak besar pada efisiensi, produktivitas, dan kualitas hidup manusia. Teknologi menawarkan berbagai solusi inovatif, termasuk salah satunya penggunaan analisis wajah untuk mendeteksi tanda - tanda autisme (Brynjolfsson & McElheran, 2016; Kasyanov, 2020).

Deteksi wajah autisme adalah pendekatan teknologi yang memanfaatkan analisis wajah untuk membantu dalam diagnosis gangguan spektrum autisme. Teknologi ini sedang berkembang pesat dan memiliki akurasi dan efisiensi dalam mendeteksi autisme. Dalam konteks deteksi wajah pada autisme, teknologi berfokus pada bagaimana individu mengekspresikan emosi melalui wajah mereka. Awalnya teknologi ini hanya mengandalkan metode berbasis fitur yang menggunakan algoritma untuk mendeteksi elemen wajah seperti mata, hidung dan mulut. Metode ini memerlukan ekstraksi fitur manual dan sering mengalami kesulitan dalam menghadapi variasi dan *noise* dalam gambar. Namun, dengan kemunculan teknologi pembelajaran mesin, terutama *deep learning*, kemampuan

deteksi wajah telah mengalami perubahan yang lebih baik (Cohn & De la Torre, 2015; Krizhevsky, Sutskever & Hinton, 2012).

Deep learning merupakan suatu metode unggulan dalam mendeteksi wajah. Teknologi ini menggunakan jaringan saraf tiruan dengan banyak lapisan (*deep neural networks*) dan secara otomatis mempelajari fitur dari data gambar tanpa memerlukan ekstraksi fitur manual. *Deep learning* secara signifikan telah meningkatkan akurasi deteksi wajah dibandingkan dengan metode tradisional. *Deep learning* telah menunjukkan kemampuan untuk mencapai tingkat ketepatan yang sangat tinggi pada deteksi wajah, bahkan dalam kondisi pencahayaan yang buruk, sudut pandang yang ekstrem, atau ekspresi wajah yang berubah (Krizhevsky, Sutskever & Hinton, 2012).

CNN (*Convolutional Neural Network*) merupakan salah satu jenis jaringan saraf dari *deep learning* yang sangat efektif untuk menganalisis data yang memiliki struktur grid, seperti gambar. CNN telah menjadi metode standar dalam pemrosesan gambar dan video, ini dikarenakan kemampuannya dalam mengidentifikasi pola gambar dan video. Pada penelitian ini, penulis menggunakan CNN sebagai metode yang dapat menunjang keberhasilan dari penelitian yang akan dilakukan. Berkat struktur konvolusi dan *pooling*-nya, CNN mampu mengenali objek, CNN dapat mengenali objek meskipun ada perubahan kecil dalam posisi atau skala. Selain itu, CNN telah menunjukkan hasil yang unggul dalam berbagai tugas pengolahan gambar, seperti klasifikasi gambar, deteksi objek, dan segmentasi. Oleh karena itu, CNN berpotensi menetapkan diagnosis autisme pada anak secara akurat (Liu & Yi, 2016).

CNN juga terdapat beberapa pemodelan salah satunya VGG16 yang merupakan arsitektur CNN yang banyak digunakan dalam deteksi wajah dan berbagai aplikasi pengenalan gambar. VGG16 telah menjadi salah satu arsitektur CNN paling populer karena kesederhanaannya dan efektivitasnya dalam berbagai tugas visi komputer. Penulis memutuskan menggunakan VGG16 sebagai pemodelan dalam penelitian ini. Hal ini karena VGG16 digunakan secara luas dalam deteksi wajah karena kemampuannya untuk menangkap fitur visual yang kompleks dan tingkat detail yang tinggi. Sehingga VGG16 dapat menjadi dasar yang kuat untuk deteksi wajah autisme jika di-fine-tune dengan dataset yang relevan dan spesifik (Krizhevsky, Sutskever & Hinton, 2012).

Penelitian ini bertujuan untuk mengeksplorasi deteksi dan klasifikasi autisme dengan menggunakan teknologi *deep learning*, khususnya dalam mengidentifikasi jenis - jenis autisme yang ditandai oleh ciri-ciri fisik tertentu. Dalam spektrum autisme, tidak semua individu menunjukkan gejala yang sama secara fisik maupun perilaku, namun terdapat subtype yang memiliki karakteristik fisik yang lebih jelas, seperti wajah atau postur yang khas, yang mungkin terkait dengan kondisi sindrom tertentu. Salah satu fokus dalam penelitian ini adalah mengidentifikasi autisme yang memiliki fitur fisik seperti karakteristik wajah mongoloid, yang mungkin dapat memberikan petunjuk tambahan dalam diagnosis dini atau penanganan autisme. Dengan demikian, penelitian ini diharapkan dapat memberikan sumbangan dalam pemahaman lebih lanjut mengenai pentingnya faktor fisik dalam mendeteksi dan mendiagnosis autisme, serta bagaimana

teknologi modern seperti *deep learning* dapat membantu proses ini secara lebih efisien dan akurat (Gosselin & Schyns, 2020).

Dalam penelitian ini, penulis juga berfokus pada akurasi deteksi wajah untuk diagnosis autisme menggunakan teknologi *deep learning*, khususnya CNN. Berbeda dengan penelitian sebelumnya yang sering mengandalkan metode berbasis fitur manual yang terbatas dalam menghadapi variasi dan *noise* gambar. Keunggulan utama dari penelitian ini terletak pada kemampuan CNN untuk mengenali pola wajah secara mendetail dengan akurasi tinggi dan memungkinkan deteksi autisme yang lebih presisi dan konsisten. Selain itu, pendekatan ini meningkatkan efisiensi dan efektivitas dalam proses diagnosis dibandingkan dengan metode deteksi wajah tradisional. Metode tradisional sering kali memerlukan penyesuaian manual dan kesulitan dalam menangani kompleksitas data wajah, sedangkan teknologi CNN dapat mengotomatisasi proses ini dengan lebih baik, mengurangi kebutuhan akan intervensi manual dan meningkatkan akurasi deteksi (LeCun, Bengio & Hinton, 2015).

Dengan pernyataan tersebut, maka penulis ingin melakukan penelitian dengan judul **“Pengukuran Tingkat Akurasi Deteksi Wajah Anak Autis Menggunakan CNN (*Convolutional Neural Network*)”**.

1.2 Rumusan Masalah

Berdasarkan identifikasi permasalahan yang telah disajikan, rumusan masalah dalam penelitian ini adalah bagaimana mengukur tingkat akurasi deteksi wajah anak autis menggunakan VGG16?

1.3 Tujuan Penelitian

Adapun tujuan penelitian yang dilakukan oleh penulis adalah untuk mengetahui tingkat akurasi deteksi wajah anak autis menggunakan model klasifikasi VGG16.

1.4 Batasan Penelitian

Adapun Batasan masalah diantaranya sebagai berikut:

1. Deteksi autisme dan klasifikasi gambar wajah menggunakan CNN.
2. Pengukuran deteksi wajah anak autis ini hanya menggunakan, Linux dan Vscod.
3. Kaggle menjadi salah satu acuan untuk mengumpulkan gambar wajah dari kelompok anak - anak yang heterogen.

1.5 Manfaat Penelitian

Adapun manfaat dari pelaksanaan penelitian diantaranya sebagai berikut:

1. Bagi mahasiswa :
 - a) Menjadikan hasil penelitian yang dilakukan oleh penulis sebagai referensi untuk mengetahui pengukuran tingkat akurasi deteksi wajah anak autis berbasis AI. - R A N I R Y
 - b) Menerapkan ilmu dan teori yang telah didapatkan pada perkuliahan.
 - c) Meningkatkan kreativitas dengan memanfaatkan teknologi komputer secara nyata menggunakan *database*.
2. Bagi Instansi/ Unit Profesi:
 - 1) Memudahkan mereka dalam melakukan diagnosis suatu penyakit dengan tingkat keakuratan yang terjamin.

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Penelitian terdahulu merujuk pada kajian-kajian ilmiah yang telah dilakukan sebelum penelitian yang sedang dilakukan saat ini. Penelitian ini memberikan latar belakang, konteks, dan dasar untuk studi baru dengan mengidentifikasi temuan - temuan sebelumnya, metodologi yang digunakan, serta batasan dan kesenjangan dalam pengetahuan yang ada. Berikut beberapa penelitian sebelumnya yang membahas mengenai penelitian yang penulis lakukan.

Penelitian yang pertama dilakukan oleh (Israr Ahmad dkk, 2024) yang berjudul “Deteksi Gangguan Spektrum Autisme Menggunakan Citra Wajah”. Penelitian ini bertujuan melakukan kinerja jaringan saraf konvolusional yang telah dilatih sebelumnya seperti ResNet34, ResNet50, AlexNet, MobileNetV2, VGG16, dan VGG19 untuk mendiagnosis ASD dan membandingkan kinerjanya. *Dataset* yang digunakan pada penelitian ini sebanyak 2.940 citra dan terbagi menjadi 2 kelas yaitu *autism* dan *non-autism*. *Dataset* disimpan kedalam dua pengaturan yang berbeda, pada pengaturan pertama dataset disimpan dalam 124×124 piksel untuk masing-masing, dan yang kedua 248×248 piksel untuk dilakukan setiap gambar meniru pencarian grid sebagai penyelidikan hubungan antara kinerja dan ukuran piksel gambar.

Penelitian selanjutnya dilakukan oleh (Irfan Maulana, Nabila Khairunnisa, Ratna Mufidah, 2023) yang berjudul “Deteksi Bentuk Wajah Menggunakan *Convolutional Neural Network* (CNN)”. Penelitian ini bertujuan untuk

membedakan bentuk wajah dengan memanfaatkan sistem saraf konvolusional (CNN). Studi ini dibuat untuk memberikan pemahaman yang lebih mendalam tentang titik fokus dan hambatan strategi CNN dalam menangani variasi kompleks pada gambar wajah. Pada penelitian ini menggunakan *dataset* 5000 gambar, dengan setiap kategori berisi 1000 gambar. Dari jumlah tersebut, 800 gambar digunakan untuk pelatihan (*training*) dan 200 gambar lainnya untuk validasi. Proses pelatihan dilakukan dengan menggunakan 10 hingga 40 epoch, menghasilkan akurasi sebesar 74%.

Penelitian selanjutnya dilakukan oleh (Fadhila Purboningrum, Fitri Utaminingrum, Fais Al Huda, 2024) yang berjudul “Sistem Deteksi Dini *Autism Spectrum Disorder* (ASD) Berbasis *Face Recognition* Menggunakan Metode *Transfer Learning* Resnet50”. Penelitian ini bertujuan untuk implementasi aplikasi *mobile* android yang nantinya akan menjadi sebuah *system* yang dapat digunakan oleh pengguna dalam melakukan deteksi dan klasifikasi *autism*. Pada penelitian ini *dataset* yang didapatkan akan dilakukan pembagian yaitu 80:10:10 untuk *Training: Validation:Test* pada masing - masing kelas *autistic* dan *non-autistic* dengan menggunakan pemodelan Resnet50. Proses klasifikasi melibatkan pembacaan data citra wajah anak dengan *resize* menjadi 224 x 224 piksel, dan mendapatkan nilai rata - rata skor SUS 85,41.

Penelitian selanjutnya dilakukan oleh (Deva Ega Marinda, Imam Husni Al Amin, 2023) yang berjudul “Implementasi Metode *Convolutional Neural Network* untuk Deteksi Penggunaan Masker secara *Real-Time*”. Tujuan penelitian ini Pertimbangan ini bertujuan untuk membantu menjamin kepatuhan terbuka terhadap

ketentuan penggunaan perlindungan dan mengendalikan penyebaran infeksi COVID-19. Dalam penelitian ini sendiri menggunakan *dataset* dengan 1.300 citra bermasker dan citra *non-masker*, dan mencapai hasil 97.5% dengan informasi gambar umumnya secara efektif diidentifikasi sebanyak 78 sampel dari total 80 sampel, Pengujian ini dilakukan dengan menggunakan kamera terpisah pada soal dengan jarak 1 sampai 10 meter. Penelitian terdahulu dapat diartikan sebagai bagaimana temuan dan kontribusi penelitian yang sudah ada, penelitian terdahulu dapat dilihat pada tabel 2.1.

Tabel 2.1 Penelitian Terdahulu

Peneliti/ Tahun	Judul Penelitian	Hasil
(Israr Ahmad, Javed Rashid, Muhammad Faheem, Arslan Akram, Nafees Ahmad Khan, Riaz ul Amin, 2024)	Deteksi Gangguan Spektrum Autisme Menggunakan Citra Wajah	Hasil akurasi Model ResNet50 dengan resolusi 248×248 berhasil mencatatkan akurasi lebih dari 92%, mengungguli semua model lainnya. Di posisi kedua, model ResNet34 menunjukkan performa terbaik kedua dengan akurasi 86% pada resolusi yang sama. Sementara itu, model VGG16 dan VGG19 hanya mencapai akurasi sebesar 80% dan 86%. Model AlexNet dan MobileNetV2

		<p>mempertahankan performa sebelumnya, dengan akurasi masing-masing 74% dan 84%. Ketika diuji menggunakan resolusi 124×124, model ResNet34 dan ResNet50 mencatatkan akurasi masing-masing 83% dan 87%. Dari hasil-hasil ini, dapat disimpulkan bahwa ResNet50 memiliki kinerja yang lebih baik dibandingkan model-model lainnya.</p>
(Irfan Maulana, Nabila Khairunnisa, Ratna Mufidah, 2023)	<p>Deteksi Bentuk Wajah Menggunakan <i>Convolutional Neural Network (CNN)</i></p>	<p>Hasil penelitian yang dilakukan dengan menggunakan 40 <i>epoch</i> dan batch size 16, memperoleh hasil akurasi persiapan tertinggi sebesar 74%.</p>
(Fadhila Purboningrum, Fitri Utaminingrum, Fais Al Huda, 2024)	<p>Sistem Deteksi Dini <i>Autism Spectrum Disorder (ASD)</i> Berbasis <i>Face Recognition</i> Menggunakan Metode</p>	<p>Hasil penelitian pada system aplikasi mampu mendeteksi dan mengklasifikasikan dengan akurasi 100% pada kelas <i>autistic</i> dan 60% pada kelas <i>non-</i></p>

	<i>Transfer Learning</i> Resnet50	<i>autistic</i> . Rata – rata akurasi system adalah 80% dari total 10 kali pengujian, dengan hasil sesuai pada 8 data citra wajah.
(Deva Ega Marinda, Imam Husni Al Amin, 2023)	Implementasi Metode <i>Convolutional Neural Network</i> untuk Deteksi Penggunaan Masker secara <i>Real-Time</i>	Hasil penelitian deteksi masker secara <i>real-time</i> mencapai akurasi 97.5% dengan menggunakan kamera untuk jarak 1 hingga 10 meter.

2.2 Kajian Teoritis

2.2.1 Pengertian Autisme

Autisme adalah kekacauan formatif yang memengaruhi komunikasi, interaksi sosial, dan perilaku seseorang. autisme termasuk dalam kategori gangguan *neurodevelopmental* dan ditandai dengan pola perilaku yang terbatas dan berulang serta kesulitan dalam berinteraksi sosial dan komunikasi. Autisme dapat memiliki spektrum yang luas, dari gejala ringan hingga berat, dan dapat mempengaruhi individu dengan cara yang berbeda (American Psychiatric Association 2013).

2.2.2 Deteksi Wajah dan Teknik pengolahan Citra

Deteksi wajah adalah proses untuk mengidentifikasi dan menemukan wajah manusia dalam gambar atau video. Teknik ini digunakan dalam berbagai aplikasi seperti pengenalan wajah, interaksi manusia-komputer, dan sistem keamanan. Pengolahan citra adalah proses analisis dan manipulasi gambar digital untuk

meningkatkan kualitas atau mengekstrak informasi. Teknik deteksi wajah sering menggunakan algoritma berbasis pengolahan citra dan pembelajaran mesin untuk mengidentifikasi fitur wajah dan membedakannya dari latar belakang (Masi, Chen & Chang, 2022; Szeleski, 2022).

2.2.3 Deep Learning

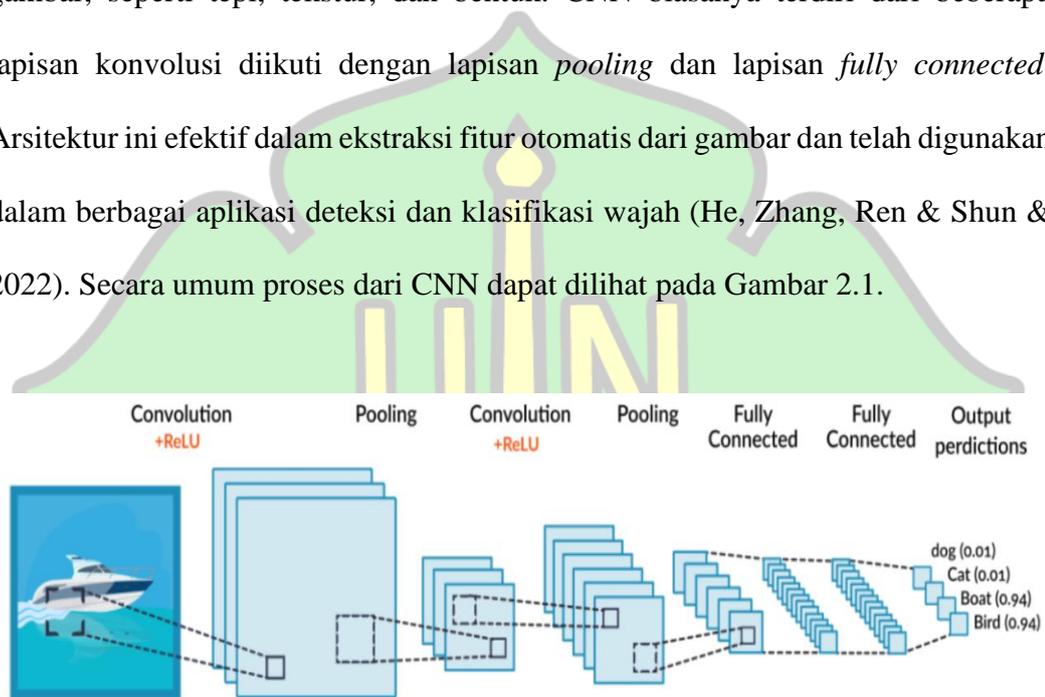
Deep learning adalah struktur jaringan saraf yang dalam (*deep neural networks*) untuk memproses dan menganalisis data. Teknik ini mampu mengidentifikasi pola kompleks dalam data melalui berbagai lapisan neuron yang saling terhubung. *Deep Learning* telah menunjukkan keberhasilan signifikan dalam berbagai aplikasi, termasuk pengolahan citra, pemrosesan bahasa alami, dan pengenalan suara (He, Zhang, Ren & Shun & 2022).

2.2.4 VGG

VGG adalah sebuah arsitektur CNN yang terkenal karena kesederhanaannya dan kemampuannya untuk mencapai hasil yang sangat baik dalam berbagai tugas pengolahan citra. VGG menggunakan lapisan konvolusi dengan filter berukuran 3x3 dan *stride* 1, yang konsisten di seluruh arsitektur. Lapisan konvolusi ini diikuti oleh lapisan pooling (biasanya *max pooling* dengan ukuran 2x2) untuk mengurangi dimensi. VGG-16 dan VGG-19 adalah dua versi yang paling umum digunakan. Angka 16 dan 19 merujuk pada jumlah total lapisan yang memiliki parameter (konvolusi dan *fully connected*).

2.2.5 CNN

Convolutional Neural Network (CNN) adalah salah satu arsitektur *deep learning* yang paling umum digunakan untuk pengolahan citra. CNN memanfaatkan lapisan konvolusi untuk mendeteksi fitur - fitur penting dalam gambar, seperti tepi, tekstur, dan bentuk. CNN biasanya terdiri dari beberapa lapisan konvolusi diikuti dengan lapisan *pooling* dan lapisan *fully connected*. Arsitektur ini efektif dalam ekstraksi fitur otomatis dari gambar dan telah digunakan dalam berbagai aplikasi deteksi dan klasifikasi wajah (He, Zhang, Ren & Shun & 2022). Secara umum proses dari CNN dapat dilihat pada Gambar 2.1.



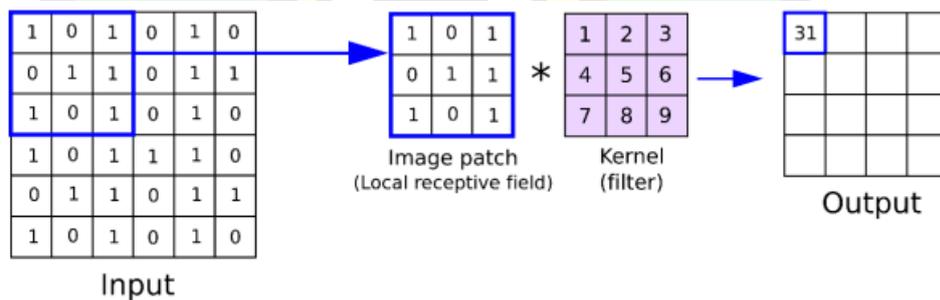
Gambar 2.1 Arsitektur CNN

A. Convolutional Layer

Convolutional Layer merupakan lapisan yang paling terorganisir dalam CNN, yang terdiri dari sejumlah saluran yang diinisialisasi secara subjektif untuk melakukan operasi konvolusi, yang berfungsi sebagai ekstraktor penyertaan untuk menemukan sorotan representasi dari gambar masukan. Nilai awal yang tidak teratur ini akan ditingkatkan di tengah pegangan persiapan. Dalam pegangan ini, bagian gambar ditingkatkan dan bergerak dari sudut atas yang kosong ke kanan

bawah. Setiap gambar kecil yang diambil dari hasil konvolusi akan digunakan sebagai input untuk membuat representasi tambahan, sehingga CNN dapat mengenali pertanyaan di mana pun pertanyaan tersebut muncul dalam gambar.

Pengukuran spasial volume hasil lapisan konvolusional adalah *hiperparameter* yang digunakan, khususnya ukuran volume, saluran, jalan, dan jumlah bantalan nol yang digunakan. Jalan digunakan untuk menggeser saluran melalui gambar masukan dan untuk mendorong angka nol di sekitar tepi gambar menggunakan bantalan nol. Dalam lapisan konvolusi, *neuron-neuron* disusun dalam suatu garis besar. Setiap *neuron* dalam garis besar tersebut seperti lapangan terbuka yang dikaitkan dengan *neuron-neuron* dalam lapisan konvolusi sebelumnya melalui serangkaian bobot yang telah disiapkan, yang juga dikenal sebagai saluran atau bit (Muhammad & Wibowo, 2021). Contoh dari ilustrasi *convolution layer* dapat dilihat pada Gambar 2.2.

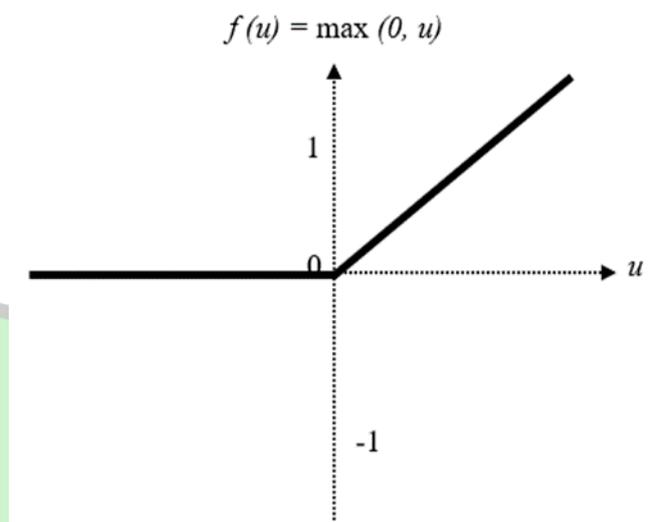


Gambar 2.2 Convolutional Layer

B. Rectified Linear Unit

Rectified Linear Unit (ReLU) adalah aktivasi yang menghasilkan nilai nol ketika $x < 0$ dan nilai linier dengan kemiringan 1 ketika $x > 0$. Aktivasi berfungsi untuk dapat bekerja setelah melakukan proses konvolusi dan mendapatkan fitur

pada suatu citra. Aktivasi ini bertujuan untuk mengubah nilai negatif pada citra menjadi nol (Abadi & Wibowo, 2021). Untuk fungsi aktivitas *Relu Layer* sendiri dapat dilihat pada Gambar 2.3.

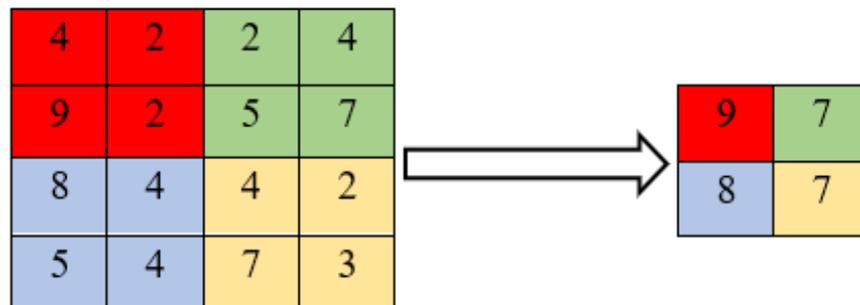


Gambar 2.3 ReLU

C. Pooling Layer

Pooling layer bisa menjadi sebuah *handle* yang bertujuan untuk mengecilkan ukuran dari sebuah informasi gambar sehingga sorotan yang akan diambil memiliki karakteristik sorotan yang jelas. Pada CNN biasanya menggunakan *subsampling max pooling*. Cara kerja *max pooling* adalah dengan memisahkan hasil dari tahap *convolution layer* yang dipersiapkan menjadi beberapa bagian kemudian mengambil nilai tertinggi dari setiap bagian kerangkanya untuk menyusun kerangka tersebut. Biasanya dilakukan untuk mengurangi tumpukan komputasi di tengah periode persiapan dan menawarkan bantuan pada kerangka tersebut jika terjadi perubahan posisi gambar. Ada dua cara untuk menentukan *pooling layer*, yaitu nilai perolehan tertinggi dan nilai perolehan rata-rata. (Abadi

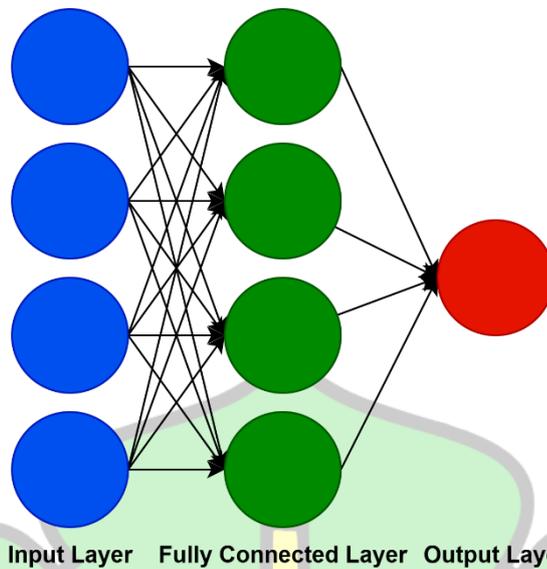
& Wibowo, 2021). Sub gambar yang digunakan dalam kegiatan ini adalah 8x8 yang dapat dilihat pada Gambar 2.4.



Gambar 2.4 *Pooling Layer*

D. *Fully Connected Layer*

Connected layer merupakan perceptron multilapis konvensional yang menggunakan kerja aktuasi *softmax* atau *sigmoid* dalam lapisan luar. Dengan "*Fully Connected*" menyatakan bahwa setiap *neuron* dalam lapisan sebelumnya dikaitkan dengan setiap neuron dalam lapisan berikutnya. Hasil dari lapisan konvolusi dan penggabungan menunjukkan fitur tingkat tinggi yang diambil dari gambar masukan. Lapisan ini pada dasarnya mengambil volume masukan (konvolusi, ReLU atau lapisan penggabungan) dan menghasilkan vektor nilai N di mana N adalah jumlah kelas. Tujuannya adalah untuk menggunakan fitur ini untuk mengklasifikasikan gambar ke dalam kelas yang berbeda berdasarkan apa yang diinisialisasi dalam data pemrosesan. (Muhammad & Wibowo, 2021). Berikut *fully Connected Layer* dapat dilihat pada Gambar 2.5.



Gambar 2.5 Fully Connected Layer

2.2.6 Confusion Matrix

Confusion matrix merupakan suatu strategi yang digunakan untuk melakukan akurasi/kalkulasi apakah suatu tayangan berkinerja baik atau tidak dalam konsep *information mining*. Penilaian dengan *framework disarray* menghasilkan nilai *precision*, *exactness* dan *review*. (Mayadewi & Rosely, 2015). Berikut tabel *Disarray Network* yang dapat dilihat pada Tabel 2.2. (Mayadewi & Rosely, 2015). Berikut tabel *Confusion Matrix* dapat dilihat pada Tabel 2.2.

Tabel 2.2 Confusion Matrix

		Prediksi	
		0	1
Aktual	0	True Positive (TP)	False Positive (FP)
	1	False Negative (FN)	True Negative (TN)

Pada perhitungan akurasi dengan tabel *Confusion Matrix* sebagai berikut:

1. Akurasi

Akurasi adalah proporsi prediksi yang benar dari total prediksi yang dilakukan oleh model. Dalam konteks deteksi wajah, akurasi mengukur seberapa sering model berhasil mendeteksi wajah dengan benar dibandingkan dengan total jumlah gambar yang dianalisis (Chicco & Jurman, 2020). Menghitung akurasi menggunakan rumus persamaan 1.

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

2. Presisi

Presisi mengukur proporsi deteksi wajah yang benar dari semua deteksi yang dilaporkan oleh model. Ini penting untuk mengevaluasi seberapa tepat model dalam mendeteksi wajah tanpa menghasilkan deteksi yang salah (Chicco & Jurman, 2020). Menghitung presisi menggunakan rumus persamaan 2.

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

3. Recall

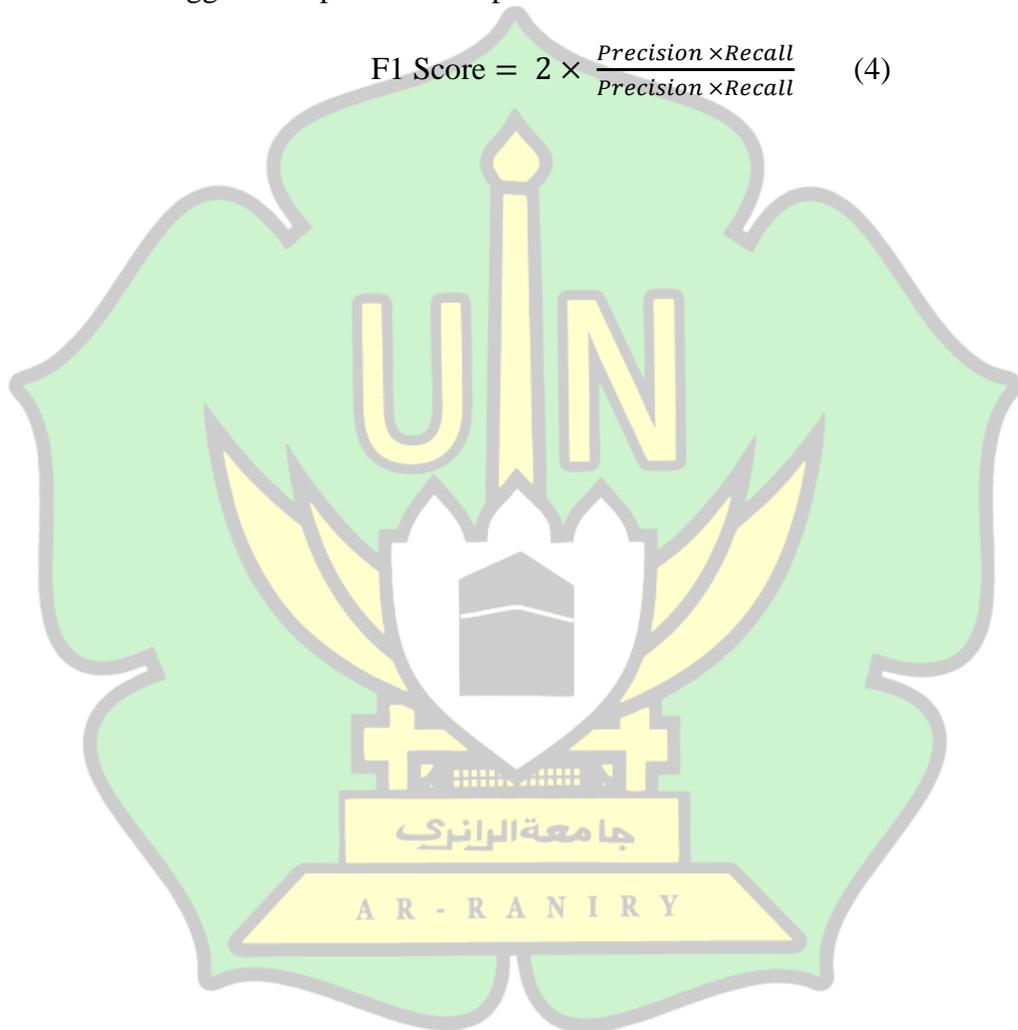
Recall mengukur proporsi wajah yang benar-benar ada yang berhasil dideteksi oleh model dari semua wajah yang ada dalam citra. Ini penting untuk memastikan bahwa model dapat menangkap sebanyak mungkin wajah yang relevan (Chicco & Jurman, 2020). *Recall* menggunakan rumus seperti persamaan 3

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

4. *F1-Score*

F1-Score adalah ukuran yang menggabungkan presisi dan *recall* menjadi satu nilai. Ini berguna untuk menilai keseimbangan antara presisi dan *recall* dalam deteksi wajah (Chicco & Jurman, 2020). Untuk menghitung ketepatan penilaian, Anda akan menggunakan persamaan seperti kondisi 4.

$$F1 \text{ Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (4)$$

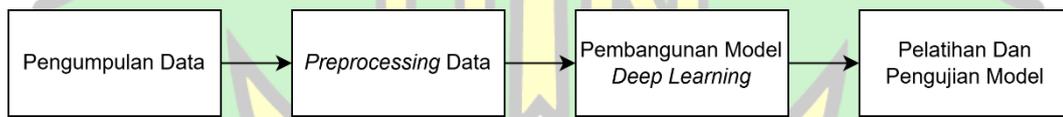


BAB III

METODE PENELITIAN

3.1 Tahapan Penelitian

Tahapan penelitian dilakukan untuk dapat mengidentifikasi dan mendefinisikan masalah yang akan diteliti, serta menetapkan tujuan penelitian yang spesifik. tahapan ini bertujuan untuk memahami konteks dan kebutuhan penelitian, serta menentukan fokus utama yaitu pengukuran tingkat akurasi deteksi wajah autisme menggunakan CNN. Tahapan penelitian diuraikan dengan jelas melalui gambar 3.1.



Gambar 3.1 Tahapan Penelitian

3.2 Pengumpulan Data

Dataset yang digunakan dalam penelitian ini diperoleh dari database wajah autisme yang diakses melalui Kaggle. Data latih (*train*) yang digunakan berjumlah 2.540 data, sementara data uji (*testing*) terdiri dari 300 data yang diambil secara acak (*shuffle*). Data yang *valid* terdiri dari 50% data *autism* dan 50% data *non-autism*, dengan masing - masing kelas memiliki 1.270 *dataset*.

Gambar wajah yang dikumpulkan harus memenuhi kriteria dari segi resolusi dimana gambar harus memiliki resolusi minimum yang ditentukan untuk memastikan kualitas pengenalan fitur. Kemudian variasi ekspresi, dimana gambar harus mencakup variasi ekspresi wajah untuk meningkatkan generalisasi model.

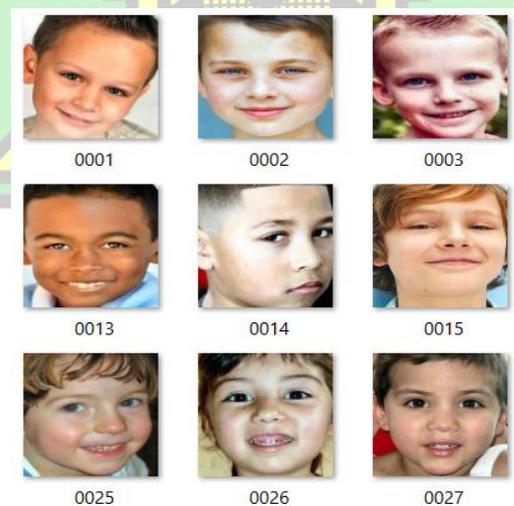
3.2.1 Jenis dan Sumber Data

Jenis data yang digunakan dalam penelitian ini adalah data sekunder, data tersebut diperoleh dengan cara mengambil data melalui Kaggle. Terdapat dua kelas dalam penelitian ini, kelas *autism* dengan 1270 data dan *non-autism* 1270 data dengan format JPG/JPEG (*Join Photographic Experts Group*). Contoh data yang didapat terletak pada gambar 3.2 dan 3.3.

Cihan Senol: <https://www.kaggle.com/datasets/cihan063/autism-image-data>



Gambar 3.2 Anak *Autism*



Gambar 3.3 Anak *Non-Autism*

3.2.2 Kebutuhan Perangkat Keras (*Hardware*)

Terdapat beberapa perangkat keras yang digunakan pada penelitian ini dengan spesifikasi laptop:

1. Processor: AMD Ryzen 5 5600H 4,2 GHz
2. Installed RAM: Random Access Memory 8 GB
3. VGA: NVIDIA GeForce GTX 1650

3.2.3 Kebutuhan Perangkat Lunak (*Software*)

Pada penelitian ini terdapat beberapa perangkat lunak yang digunakan yaitu:

1. Sistem Operasi: Linux
2. *Python*, merupakan bahasa pemrograman dengan karakter *interpretative* multiguna. *Python* bisa menjadi dialek pemrograman tingkat tinggi atau mudah dipahami orang dan mudah beradaptasi serta memiliki manajemen memori terprogram. *Python* dapat digunakan dalam berbagai aplikasi, mulai dari ilmu data hingga pengembangan program (Ihsan, 2021).
3. *TensorFlow*

TensorFlow merupakan sistem pembelajaran mesin yang diberikan Google dan merupakan sumber terbuka untuk pengembangan komputer yang cerdas. Pada *TensorFlow* tersedia antarmuka untuk mendefinisikan kalkulasi pembelajaran mesin dan dapat berjalan pada berbagai kerangka kerja. *TensorFlow* juga menjadi *library* yang paling banyak digunakan dalam ilmu informasi. (Ihsan, 2021).

4. *Library NumPy*

NumPy merupakan *library* dalam *Python* yang ditujukan untuk perhitungan logis dan ilmiah (Retnoningsih & Pramudita, 2020).

5. *Library Matplotlib*

Matplotlib merupakan *library* yang akan dimanfaatkan untuk membuat bagan plot guna membayangkan kejadian yang terjadi. (Retnoningsih & Pramudita, 2020).

6. *Library Keras*

Keras merupakan *library neural network* tingkat tinggi yang disusun dalam bahasa *python*. Keras dapat dijalankan pada sistem wawasan buatan seperti CNTK, Theano atau *TensorFlow*. Keras menyediakan fitur-fitur yang digunakan untuk mendorong pengembangan Pembelajaran Mendalam. (Santoso & Ariyanto, 2018).

7. *Open CV*

Open CV adalah pustaka perangkat lunak yang digunakan untuk memproses gambar dan video secara *real-time*.

8. *Seaborn*

Seaborn adalah pustaka visualisasi data yang dibangun di atas Matplotlib dan dirancang untuk membuat pembuatan grafik statistik yang lebih mudah dan lebih menarik.

9. *Scikit-learn*

Scikit-learn adalah pustaka *Python* yang digunakan untuk analisis data dan pemodelan mesin belajar (*machine learning*). *Scikit-learn* menyediakan

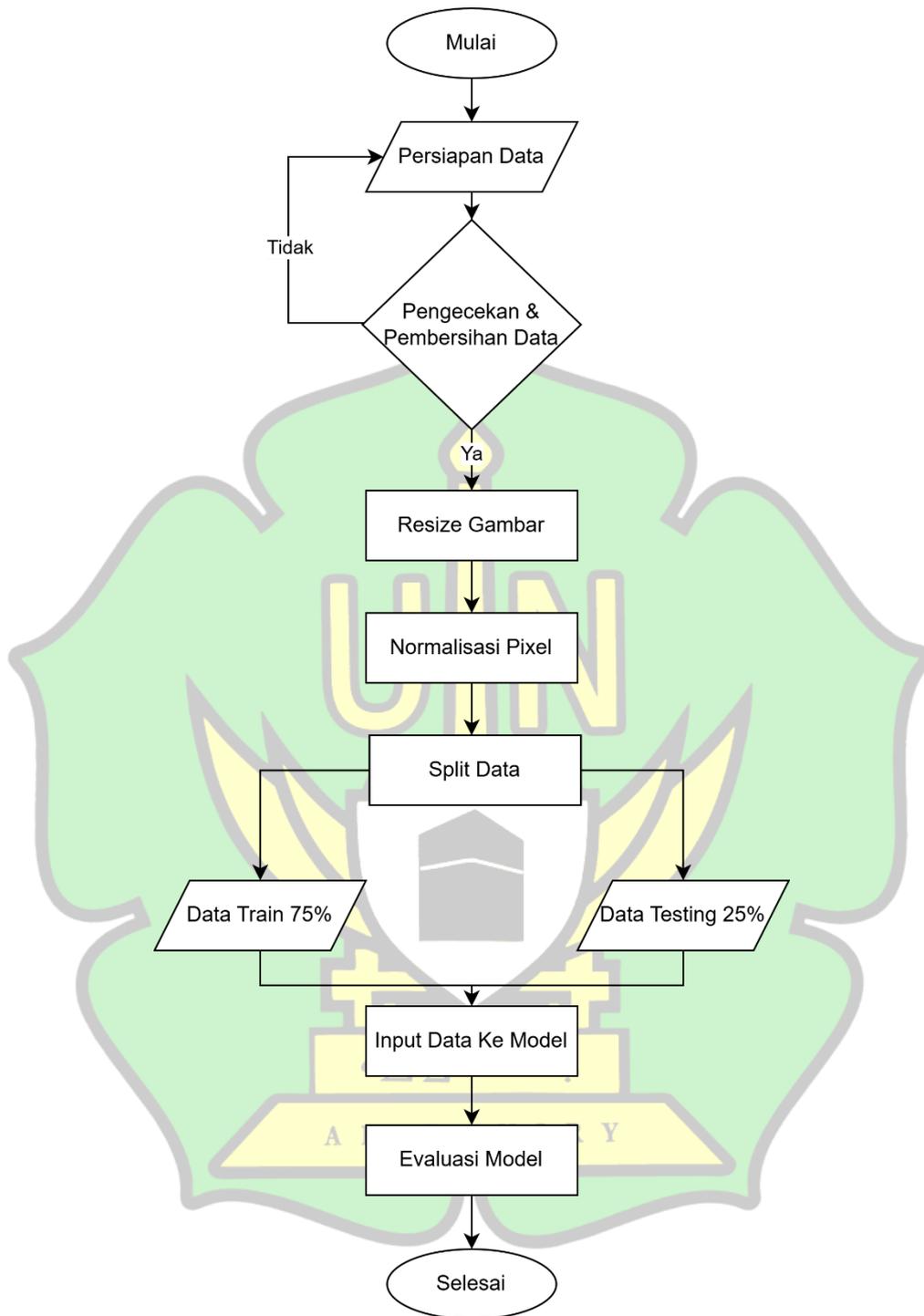
berbagai alat dan algoritma untuk klasifikasi, regresi, klusterisasi, reduksi dimensi, pemilihan fitur, dan evaluasi model.

10. VsCode

Merupakan editor kode sumber yang dikembangkan oleh Microsoft. Ini mendukung berbagai bahasa pemrograman melalui eksistensi (Santosa, 2020).

3.3 *Preprocessing Data*

Langkah - langkah *preprocessing* mencakup, menghapus gambar yang tidak jelas atau buram, mengubah ukuran gambar ke dimensi yang konsisten untuk memudahkan pemrosesan oleh model *deep learning*, dan menggunakan teknik *split data* seperti rotasi, *zoom*, dan *flipping* untuk meningkatkan keragaman dataset dan mencegah *overfitting*. Dalam penyesuaian dan normalisasi, dilakukan langkah di antaranya, meningkatkan kualitas gambar untuk memastikan fitur wajah lebih jelas dan mengubah skala *pixel* gambar sehingga nilainya berada dalam rentang [0,1] untuk meningkatkan performa model. *Flowchart Preprocessing Data* dapat dilihat pada Gambar 3.4.



Gambar 3.4 *Flowchart Preprocessing Data*

3.2.1 Penghapusan Gambar yang Tidak Jelas atau Buram

Proses ini bertujuan untuk memastikan bahwa hanya gambar yang memiliki kualitas baik yang akan digunakan dalam pelatihan model. Gambar yang terlalu buram, kabur, atau tidak memiliki fitur wajah yang jelas akan dihapus dari *dataset*. Gambar dengan resolusi rendah atau kualitas gambar yang buruk dapat mengganggu kemampuan model dalam mengenali pola atau fitur wajah. Oleh karena itu, gambar yang tidak memenuhi standar kualitas visual akan dikeluarkan dari dataset untuk menjaga keakuratan dan konsistensi model.

3.3.2 Mengubah Ukuran Gambar

Semua gambar akan diubah ukurannya ke dimensi yang sama untuk memudahkan pemrosesan oleh model *deep learning*. Gambar wajah biasanya diubah ukurannya menjadi ukuran standar, misalnya 150 x 150 pixel, agar model dapat menganalisis gambar dengan lebih efisien. Ukuran gambar yang seragam membantu jaringan saraf tiruan untuk memproses input tanpa mengalami kesulitan karena variasi dimensi.

3.3.3 Normalisasi Gambar

Normalisasi adalah proses mengubah skala nilai *pixel* gambar menjadi nilai yang berada dalam rentang tertentu, biasanya antara 0 dan 1, untuk meningkatkan performa model. Gambar digital umumnya memiliki nilai piksel dalam rentang 0 hingga 255. Dalam proses normalisasi, nilai ini disesuaikan agar setiap piksel memiliki nilai yang lebih kecil, misalnya antara 0 dan 1. Ini membantu model *deep learning* bekerja lebih efisien, karena jaringan saraf tiruan cenderung bekerja lebih

baik dengan data yang sudah dinormalisasi, menghindari nilai yang terlalu besar atau terlalu kecil.

3.3.4 *Split Data*

Split data merupakan proses untuk membagi antara data latih dan data uji. Metode *holdout validation* dan *k-fold cross validation* dapat digunakan untuk membagi data latih dan data uji. Proses validasi sangat penting untuk dilakukan, tujuannya agar setiap data memiliki peluang sebagai pelatihan data dan pengujian data. *Split data* bertujuan untuk memastikan model yang dibuat akurat dan dapat digunakan untuk proses lanjutan

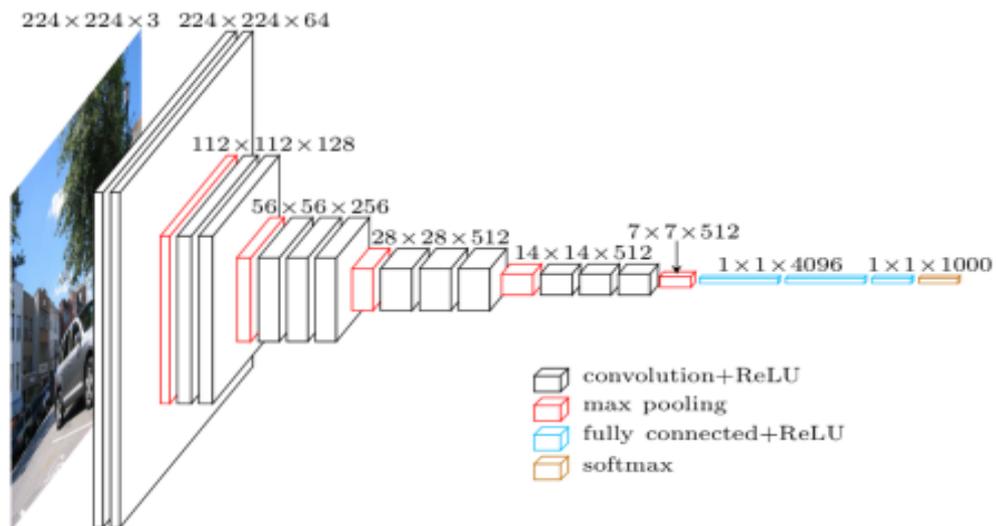
3.3.5 Peningkatan Kualitas Gambar

Proses ini melibatkan teknik-teknik tertentu untuk memperjelas fitur wajah yang ada dalam gambar, seperti kontras, kecerahan, atau ketajaman, agar lebih mudah dikenali oleh model. Gambar yang terlalu gelap atau terlalu terang mungkin menyulitkan model untuk mendeteksi fitur wajah. Oleh karena itu, peningkatan kualitas gambar dengan menyesuaikan kontras dan kecerahan dilakukan untuk memastikan bahwa detail penting seperti mata, hidung, dan mulut terlihat dengan jelas. Proses ini akan meningkatkan kemampuan model dalam mengenali wajah.

3.4 Pembuatan Model *Deep Learning*

Model *deep learning* yang digunakan dalam penelitian ini meliputi arsitektur utama yaitu VGG-16 yang merupakan model CNN yang dikenal dengan kedalaman arsitekturnya dan kemampuannya dalam ekstraksi fitur. Arsitektur

VGG-16 akan digunakan untuk ekstraksi fitur wajah. Berikut ini adalah Arsitektur VGG-16 yang dapat dilihat pada Gambar 3.5.



Gambar 3.5 Arsitektur VVG-16 (Ashushekar, 2024)

Arsitektur VGG16 merupakan model jaringan saraf konvolusional yang terdiri dari 16 lapisan utama, termasuk 13 lapisan konvolusi dan 3 lapisan *fully connected*. Model ini dirancang untuk melakukan ekstraksi fitur secara mendalam melalui serangkaian lapisan konvolusi dengan ukuran filter 3x3 dan penerapan *padding* 'same' untuk menjaga dimensi citra. Setiap blok konvolusi diikuti oleh lapisan *max-pooling* untuk mereduksi dimensi spasial dan meningkatkan invariansi terhadap transformasi citra. Setelah lapisan konvolusi dan *pooling*, data diteruskan ke tiga lapisan *fully connected* yang digunakan untuk klasifikasi, dengan lapisan output menggunakan fungsi aktivasi *softmax* untuk menghasilkan probabilitas untuk setiap kelas.

Berikut ini tahapan-tahapan penggunaan VGG-16 dalam mendeteksi tingkat akurasi pada wajah anak autisme.

3.4.1 Instalasi *Environment*

Beberapa waktu lalu dalam rangka mempersiapkan data untuk meningkatkan efektivitas pemrosesan data dalam *Convolutional Neural Network* (CNN), peneliti perlu menyiapkan lingkungan yang tepat. Karena tahap ini menggunakan bahasa pemrograman *Python* dan berfokus pada klasifikasi gambar, beberapa pustaka yang harus disiapkan antara lain NumPy, Keras, *scikit-learn*, Matplotlib, *OpenCV* dan *Stack Pre-Trained VGG-16*. Penginstalan dilakukan dengan menggunakan command prompt/terminal dengan menggunakan 'pip'.

3.4.2 *Input Gambar*

Pada tahap awal, gambar wajah anak autisme yang telah melalui *preprocessing* (misalnya, normalisasi, *split* data, dan perubahan ukuran) dimasukkan ke dalam arsitektur VGG16. Prosesnya di antaranya sebagai berikut:

1. Gambar - gambar tersebut akan berukuran 150 x 150 piksel dan terdiri dari 3 warna RGB (*Red Green Blue*).
2. Setiap gambar akan menjadi input yang diproses melalui lapisan konvolusi pertama VGG16.

3.4.3 *Convolutional Layers*

Definisi: VGG16 memiliki serangkaian lapisan proses konvolusi untuk mengekstrak fitur dari gambar *input*, seperti tepi, sudut, bentuk dasar, dan pola wajah yang lebih kompleks. Proses dari tahap ini antara lain sebagai berikut:

1. Setiap lapisan konvolusi menggunakan filter 3x3 yang bertugas untuk mendeteksi fitur kecil pada gambar. Pada lapisan awal, filter ini mendeteksi fitur-fitur dasar seperti garis, bentuk bulat, atau tekstur. Setelah melalui

beberapa lapisan, model mulai mengenali pola yang lebih kompleks seperti mata, hidung, mulut, dan ekspresi wajah.

2. Setiap lapisan konvolusi diikuti oleh fungsi aktivasi ReLU (*Rectified Linear Unit*) yang menghapus nilai negatif dan mempertahankan fitur-fitur yang relevan, membuat deteksi fitur lebih efisien.

3.4.4 *Pooling Layers*

Pooling layer bertujuan untuk mengurangi dimensi data (*subsampling*) sambil mempertahankan informasi penting yang telah diekstraksi oleh lapisan konvolusi. VGG16 menggunakan teknik *Max Pooling* untuk mendapatkan nilai maksimum dari setiap bagian kecil pada fitur yang diekstraksi. Prosesnya antara lain sebagai berikut:

1. Setelah fitur diekstraksi oleh lapisan konvolusi, data yang dihasilkan akan dilewatkan ke *pooling layer*.
2. *Pooling* ini akan mengambil nilai maksimum dari sekumpulan piksel untuk mengurangi resolusi gambar, sambil tetap mempertahankan fitur yang paling penting.
3. Ini memungkinkan jaringan untuk fokus pada elemen kunci dari wajah anak autisme, seperti ekspresi tertentu atau bentuk wajah yang khas, tanpa kehilangan banyak informasi.

3.4.5 *Fully Connected Layers*

Setelah melalui lapisan konvolusi dan *pooling*, data yang telah diproses (berupa fitur-fitur yang telah diidentifikasi) akan dilewatkan ke lapisan *fully*

connected. Di sinilah keputusan akhir dibuat berdasarkan informasi yang telah dipelajari dari gambar. Berikut proses dari tahap ini:

1. Data yang telah diekstraksi akan diflatkan menjadi vektor satu dimensi dan dilewatkan melalui beberapa lapisan *fully connected*.
2. Lapisan ini bertugas untuk mengklasifikasikan apakah gambar tersebut merupakan wajah anak dengan autisme atau tidak, berdasarkan fitur-fitur yang telah dipelajari.
3. VGG16 biasanya menggunakan fungsi aktivasi *Softmax* pada lapisan terakhir untuk menghasilkan probabilitas dari setiap kelas (autisme atau non-autisme), memungkinkan model menentukan dengan tingkat keyakinan tertentu apakah gambar tersebut mengindikasikan ciri - ciri autisme.

3.4.6 Proses Backpropagation

Backpropagation adalah metode yang digunakan untuk mengoptimalkan bobot dalam jaringan saraf tiruan, dengan menyesuaikan bobot berdasarkan kesalahan yang dihasilkan oleh prediksi model. Prosesnya adalah sebagai berikut:

1. Setelah gambar diproses dan prediksi dibuat, model akan menghitung kesalahan (*error*) antara prediksi dan label sebenarnya (*autism* atau *non-autism*).
2. Kesalahan ini kemudian digunakan untuk memperbarui bobot dalam jaringan, sehingga model dapat belajar dan memperbaiki prediksinya di masa mendatang. Proses ini diulang untuk setiap gambar dalam dataset selama beberapa *epoch* hingga model mencapai akurasi yang tinggi.

3.4.7 Evaluasi Akurasi Model

Evaluasi akurasi dilakukan setelah model selesai dilatih, menggunakan data uji yang terpisah dari data pelatihan. Model dinilai berdasarkan seberapa baik ia bisa mengklasifikasikan gambar - gambar wajah autisme yang belum pernah dilihat sebelumnya. Prosesnya adalah sebagai berikut:

1. Setelah pelatihan, data uji yang belum pernah digunakan dimasukkan ke dalam model.
2. Model akan memprediksi apakah gambar tersebut menunjukkan wajah anak dengan autisme atau tidak.
3. Akurasi dihitung sebagai perbandingan antara prediksi yang benar dan total prediksi yang dilakukan oleh model, memberikan gambaran seberapa baik VGG16 dalam mendeteksi wajah anak autisme.

3.5 Pelatihan dan Pengujian Model

Convolutional Neural Network memerlukan penanganan persiapan dan pengujian. Persiapan ini bertujuan untuk menyiapkan kalkulasi CNN guna mengenali himpunan data dan menghasilkan tampilan berdasarkan persiapan tersebut. Untuk menemukan tampilan tersebut, beberapa putaran dilakukan pada himpunan data yang disebut *usia*. Persiapan pengujian bertujuan untuk menguji tampilan yang telah dibentuk selama persiapan persiapan, persiapan persiapan, dan pengujian tampilan menggunakan data uji dari hasil persiapan gambar yang telah dilakukan sebelumnya. Susunan ini akan memberikan hasil pada data yang telah diolah menggunakan tampilan CNN yang telah dibuat.

3.5.1 Pelatihan Algoritma CNN

Pada tahap ini, algoritma CNN dilatih untuk mengenali dataset yang telah diproses. Pelatihan dilakukan dengan memberikan gambar wajah autisme dan gambar normal ke dalam jaringan CNN agar model bisa mempelajari pola dari data tersebut. Berikut proses pada pelatihan algoritma CNN:

1. Gambar-gambar yang sudah melalui tahap preprocessing dimasukkan ke dalam CNN.
2. CNN akan memproses setiap gambar melalui beberapa lapisan, seperti lapisan konvolusi, *pooling*, dan *fully connected*.
3. Dalam setiap lapisan, CNN mengekstrak fitur - fitur penting dari gambar (seperti bentuk mata, mulut, atau hidung) dan memperkuat pola yang relevan untuk deteksi autisme.
4. Model dilatih dengan menggunakan teknik *backpropagation* untuk menyesuaikan bobot setiap neuron dalam jaringan, sehingga model bisa meningkatkan akurasi deteksinya. Pelatihan ini dilakukan selama beberapa kali pengulangan (*epoch*), sehingga model bisa mempelajari pola dengan lebih baik.

3.5.2 Proses Pengujian Model

Setelah pelatihan selesai, model yang telah dilatih akan diuji dengan data yang berbeda dari data yang digunakan selama pelatihan untuk melihat seberapa baik model bisa mendeteksi autisme pada gambar wajah yang belum pernah dilihat sebelumnya. Berikut proses dari pengujian model:

1. Dataset uji yang tidak digunakan saat pelatihan dimasukkan ke dalam model CNN.
2. Model akan memproses gambar-gambar tersebut, sama seperti pada tahap pelatihan, dan memberikan prediksi apakah gambar tersebut termasuk ke dalam kategori autisme atau tidak.
3. Hasil prediksi kemudian dibandingkan dengan label sebenarnya untuk menilai kinerja model, dengan mengukur metrik seperti akurasi, presisi, *recall*, dan *F1-Score*.

3.5.3 Pengolahan Data Uji

Data uji digunakan untuk mengevaluasi model yang telah dilatih. Data ini merupakan gambar yang telah melalui tahap preprocessing yang sama seperti data pelatihan, namun tidak ikut serta dalam pelatihan itu sendiri. Proses dari pengobatan data uji ini antara lain sebagai berikut:

1. Data uji diproses menggunakan model CNN yang sudah dilatih.
2. Model akan melakukan prediksi pada data ini, dan output prediksinya tidak sebanding dengan label asli dengan tujuan menghitung performa model dan akurasi pertunjukan.
3. Jika hasilnya menunjukkan bahwa model bekerja dengan baik pada data uji, maka model dapat dianggap mampu mengenali wajah autisme secara efektif.

3.5.4 Proses *Epoch*

Epoch adalah satu putaran penuh dari proses pelatihan model, di mana seluruh dataset pelatihan diproses oleh model satu kali. Beberapa *epoch* digunakan

untuk memastikan model dapat mempelajari pola dengan lebih baik. Proses dari *Epoch* antara lain sebagai berikut:

1. Dalam setiap *epoch*, model diberikan seluruh gambar dari dataset pelatihan.
2. Setelah setiap *epoch*, model akan mengupdate bobot *neuron* berdasarkan kesalahan yang ditemukan dalam prediksi (*error*).
3. Proses ini diuji beberapa kali dari 10, 20, 30 dan 40 *epoch* hingga model mencapai tingkat akurasi yang diinginkan.



BAB IV

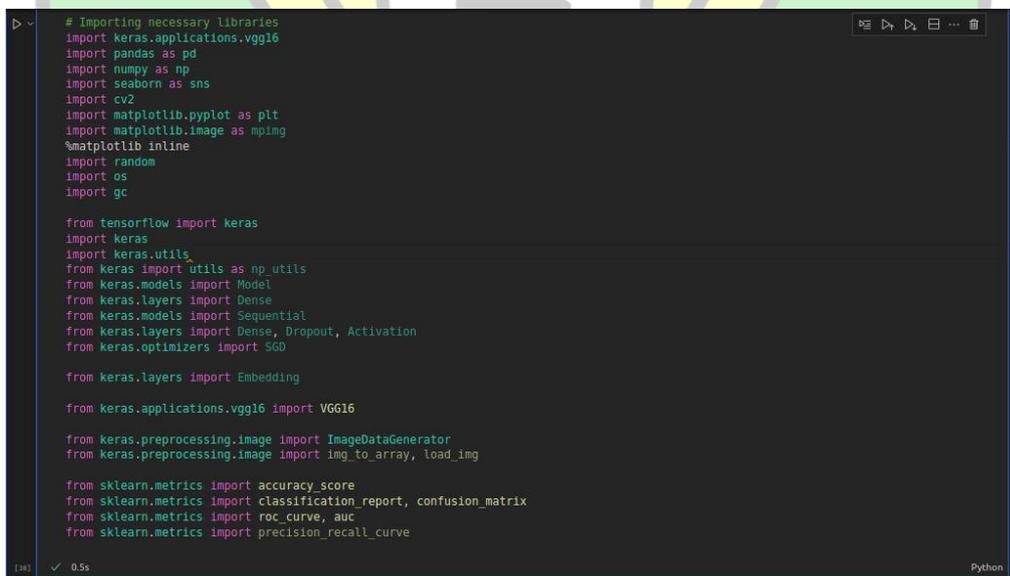
HASIL DAN PEMBAHASAN

4.1 Hasil Penelitian

4.1.1 Persiapan Data

Pada tahap awal penelitian ini adalah mempersiapkan dan mengumpulkan *dataset* yang akan digunakan untuk penelitian ini yang dikutip dari salah satu website yaitu www.kaggle.com. Data yang di ambil berjumlah 2540 data anak autis 1270 dan tidak autis 1270 yang akan di uji dengan data *testing* 25% dari jumlah *dataset* yang ditemukan.

Beberapa pemanggilan *Library Tensorflow* yang digunakan untuk penelitian ini dapat dilihat pada gambar 4.1.



```
# Importing necessary libraries
import keras.applications.vgg16
import pandas as pd
import numpy as np
import seaborn as sns
import cv2
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
%matplotlib inline
import random
import os
import gc

from tensorflow import keras
import keras
import keras.utils
from keras import utils as np_utils
from keras.models import Model
from keras.layers import Dense
from keras.models import Sequential
from keras.layers import Dense, Dropout, Activation
from keras.optimizers import SGD

from keras.layers import Embedding

from keras.applications.vgg16 import VGG16

from keras.preprocessing.image import ImageDataGenerator
from keras.preprocessing.image import img_to_array, load_img

from sklearn.metrics import accuracy_score
from sklearn.metrics import classification_report, confusion_matrix
from sklearn.metrics import roc_curve, auc
from sklearn.metrics import precision_recall_curve
```

Gambar 4.1 *Import Library*

Pada tahapan pertama yang dilakukan merupakan pemanggilan beberapa *library* termasuk *library tensorflow* tersebut yang akan digunakan dalam penelitian

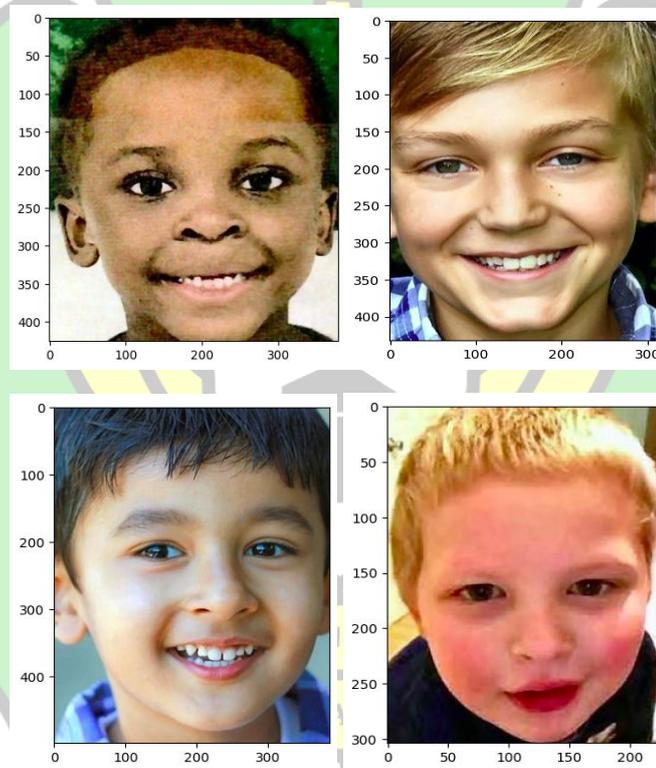
ini sebagai langkah awal dan dan kemudian kita menjalankan codingan tersebut dengan perintah pada gambar 4.2.

```
# Creating file path for our train data and test data
train_dir = "AutismDataset/train"
test_dir = "AutismDataset/test"
```

[2] ✓ 0.1s Python

Gambar 4.2 Code Membaca Data

Contoh sample data gambar yang dikutip dari *website kaggle* dapat dilihat dari gambar 4.3.



Gambar 4.3 Contoh Sample Data

4.1.2 *Autism*

Pada tahap berikut merupakan data atau gambar-gambar anak *autism* yang di ambil dari kaggle yang berjumlah 1270. Contoh gambar anak *autism* seperti pada gambar 4.4.



Gambar 4.4 Contoh Gambar *Autism*

4.1.3 *Non Autism*

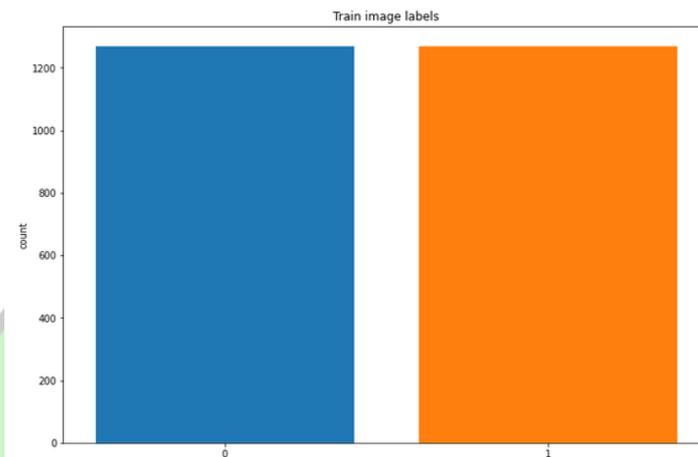
Pada tahap berikut merupakan data atau gambar-gambar anak *non autism* yang di ambil dari kaggle yang berjumlah 1270. Contoh gambar anak *non autism* seperti pada gambar 4.5.



Gambar 4.5 Contoh Gambar *Non Autism*

4.1.4 Dataset

Berikut merupakan grafik dari *dataset* yang diambil dari *website* kaggle dan dijalankan dengan jumlah data 2540. Dapat dilihat pada gambar 4.6.



Gambar 4.6 Grafik Dataset

4.1.5 Data Training

Data *training* yang digunakan sebesar dari 1270 data autism dan 1270 non autism tersebut akan di dibagi dengan jumlah data *testing* tersebut.

4.1.6 Data Testing

Data *testing* diambil dari *dataset* yang digunakan sebanyak sebanyak 300 dari banyaknya data *training* yang digunakan.

4.1.7 Model Vgg16

Tahap selanjutnya adalah memanggil model Vgg16 yang dimana untuk memanggil tahap ini dapat dilihat pada gambar 4.7.

```
# Memanggil model VGG16 terlatih
base_model = VGG16(include_top=False, weights='imagenet', input_shape=(150, 150, 3))
```

Gambar 4.7 Memanggil Model Vgg16

Tahap berikutnya yaitu menggunakan *library* Keras untuk membangun sebuah model klasifikasi menggunakan arsitektur *Sequential*. Tahap bertujuan untuk membangun model klasifikasi berbasis transfer learning menggunakan model Vgg16. Lapisan tambahan seperti *Flatten*, *Dense*, dan *Dropout* ditambahkan untuk menyempurnakan kemampuan klasifikasi sesuai dengan kebutuhan *dataset* dan masalah spesifik.

```
[17] # buat ulang model pengklasifikasi kita, sambungkan model vgg yang telah dilatih sebelumnya ke model
model = keras.models.Sequential()
model.add(base_model)
model.add(layers.Flatten())
model.add(layers.Dense(512, activation = 'relu'))
model.add(layers.Dropout(0.5))
model.add(layers.Dense(1, activation = 'sigmoid'))
Python
```

Gambar 4.8 Menyambungkan Model Vgg16

Gambar 4.9 merupakan hasil dari codingan pada gambar 4.8 yang dimana berisi tentang arsitektur model serta informasi tentang jumlah parameter dan bentuk *output* disetiap lapisan.

```
[18] # Buat ringkasan model
model.summary()
... Model: "sequential"
-----
Layer (type)                Output Shape              Param #
-----
vgg16 (Functional)          (None, 4, 4, 512)        14714688
flatten (Flatten)           (None, 8192)              0
dense (Dense)                (None, 512)               4194816
dropout (Dropout)           (None, 512)               0
dense_1 (Dense)              (None, 1)                 513
-----
Total params: 18910017 (72.14 MB)
Trainable params: 4195329 (16.00 MB)
Non-trainable params: 14714688 (56.13 MB)
```

Gambar 4.9 Hasil Model Vgg16

4.1.8 Generator Pengujian Gambar

Pada tahap ini merupakan tahap pembuatan generator data menggunakan fungsi *flow* dari *library* Keras, yang digunakan untuk menghasilkan *batch* data secara *real-time* saat pelatihan dan validasi model.

```
# Buat generator gambar pengujian dan validasi
train_generator = train_datagen.flow(X_train, y_train, batch_size = batch_size)
val_generator = val_datagen.flow(X_val, y_val, batch_size = batch_size)
```

Gambar 4.10 Generator Pengujian Gambar

Tujuan dari tahap ini adalah membuat generator untuk data pelatihan, yang akan menghasilkan *batch* data selama pelatihan untuk efisiensi memori.

4.1.9 Training Model

Tahap selanjutnya merupakan bagian dari proses pelatihan model dalam pembelajaran mesin menggunakan *TensorFlow* atau Keras. *Training* model ini sendiri dilakukan beberapa kali dengan *epoch* 10, 20, 30, dan 40.

1) *Epoch* 10

Pengujian pertama dilakukan dengan 10 *epoch* yang dimana dapat dilihat pada gambar 4.11.

```
# Train the model
history = model.fit(train_generator,
                    steps_per_epoch=ntrain // batch_size,
                    epochs=10,
                    validation_data=val_generator,
                    validation_steps=nval // batch_size)
```

Gambar 4.11 *Training* Model 10 *Epoch*

2) *Epoch* 20

Pengujian kedua dilakukan dengan 20 *epoch* yang dimana dapat dilihat pada gambar 4.12.

```
# Train the model
history = model.fit(train_generator,
                    steps_per_epoch=ntrain // batch_size,
                    epochs=20,
                    validation_data=val_generator,
                    validation_steps=nval // batch_size
                    )
```

Gambar 4.12 Training Model 20 Epoch

3) Epoch 30

Pengujian ketiga dilakukan dengan 30 epoch yang dimana dapat dilihat pada gambar 4.13.

```
# Train the model
history = model.fit(train_generator,
                    steps_per_epoch=ntrain // batch_size,
                    epochs=30,
                    validation_data=val_generator,
                    validation_steps=nval // batch_size
                    )
```

Gambar 4.13 Training Model 30 Epoch

4) Epoch 40

Pengujian ketiga dilakukan dengan 30 epoch yang dimana dapat dilihat pada gambar 4.13.

```
# Train the model
history = model.fit(train_generator,
                    steps_per_epoch=ntrain // batch_size,
                    epochs=40,
                    validation_data=val_generator,
                    validation_steps=nval // batch_size
                    )
```

Gambar 4.14 Training Model 40 Epoch

Tahap ini bertujuan untuk melatih model pembelajaran mesin menggunakan data yang dihasilkan oleh generator, yang sering digunakan untuk menangani dataset besar yang tidak dapat dimuat sepenuhnya ke dalam memori.

4.1.10 Label Prediksi Gambar

Tahap selanjutnya adalah membuat prediksi pada data uji (X) menggunakan model yang sudah dilatih dan kemudian mengonversi hasil prediksi menjadi label biner (0 atau 1) berdasarkan ambang batas tertentu (*threshold*).

```
[23] # Label prediksi untuk gambar uji
pred = model.predict(X)
threshold = 0.5
predictions = np.where(pred > threshold, 1,0)
#print(predictions)
Python
```

Gambar 4.15 Label Prediksi Gambar

Tujuan dari tahap ini adalah menghasilkan label klasifikasi yang dapat digunakan untuk analisis lebih lanjut atau evaluasi model dan mempermudah interpretasi hasil prediksi dengan mengonversi probabilitas menjadi label biner.

Berikut merupakan label gambar uji dan prediksi dapat dilihat pada gambar 4.13.

```
[24] # Plot gambar uji dan prediksi terkaitnya
test = pd.DataFrame(data = predictions, columns = ["predictions"])
test
test["filename"] = [os.path.basename(i) for i in test_imgs]
test["test_labels"] = y_test
test = test[["filename", "test_labels", "predictions"]]
test
```

...	filename	test_labels	predictions
0	Non_Autistic.15.jpg	0	1
1	Autistic.52.jpg	1	1
2	Autistic.51.jpg	1	1
3	Non_Autistic.54.jpg	0	1
4	Non_Autistic.119.jpg	0	0
...
295	Autistic.127.jpg	1	0
296	Non_Autistic.28.jpg	0	0
297	Autistic.97.jpg	1	1
298	Non_Autistic.143.jpg	0	0
299	Non_Autistic.86.jpg	0	0

300 rows x 3 columns

Gambar 4.16 Hasil Gambar Uji Dan Prediksi

4.1.11 Model Akurasi

Tahap selanjutnya adalah menghitung dan menampilkan akurasi model klasifikasi dengan membandingkan label sebenarnya (*y_test*) dengan label prediksi (*predictions*) yang dihasilkan oleh model.

```
model_accuracy = accuracy_score(y_test, predictions)
print("Model Accuracy: {:.2f}%".format(model_accuracy * 100))

[29] Python
... Model Accuracy: 50.67%
```

Gambar 4.17 Model Akurasi

Mengukur seberapa baik model dapat memprediksi label sebenarnya pada dataset uji dan memberikan metrik performa dasar yang mudah dipahami.

```
# Plot train and validation accuracy
plt.figure(figsize=(12, 8))
sns.lineplot(data=history_df.loc[:, ["acc", "val_acc"]], palette=["b", "r"], dashes=False)
sns.set_style("whitegrid")
plt.xlabel("Epochs")
plt.ylabel("Accuracy")
plt.title("Training and Validation Accuracy")

[75] ✓ 0.5s
```

Gambar 4.18 Training and Validation Accuracy

Codingan pada gambar 4.15 merupakan visualisasi *training accuracy* dan *validation accuracy* dari suatu model selama proses pelatihan. Tujuannya adalah untuk membandingkan performa model dalam pelatihan dan validasi melalui visualisasi akurasi, sehingga mempermudah analisis terhadap *overfitting* atau *underfitting*.



Gambar 4.19 Grafik Training and Validation Accuracy

Berikutnya *training loss* dan *validation loss* dari suatu model selama proses pelatihan. Tujuannya adalah untuk Memonitor apakah model bekerja dengan baik. Mendeteksi potensi *overfitting*. tahap ini berfokus pada *loss*, yang merupakan metrik kesalahan. *Loss* lebih relevan untuk analisis awal karena merupakan fungsi objektif yang dioptimasi selama pelatihan.

```
# Plot train and validation loss
plt.figure(figsize=(12, 8))
sns.lineplot(data=history_df.loc[:, ["loss", "val_loss"]], palette=['b', 'r'], dashes=False)
sns.set_style("whitegrid")
plt.xlabel("Epochs")
plt.ylabel("Loss")
plt.title("Training and Validation Loss")
```

Gambar 4.20 *Training and Validation Loss*



Gambar 4.21 Grafik *Training and Validation Loss*

4.1.12 Laporan Klasifikasi Pada Kinerja Model

Tahap selanjutnya adalah mengevaluasi kinerja model klasifikasi secara mendalam dengan laporan klasifikasi (*Classification Report*), yang mencakup metrik penting seperti *precision*, *recall*, *f1-score*, dan *support* untuk setiap kelas. Berikut hasil – hasil dari klasifikasi kinerja model dari beberapa *epoch*.

1) *Epoch* 10

Hasil dari 10 *epoch* dapat dilihat pada gambar 4.22.

```
model_accuracy = accuracy_score(y_test, predictions)
print("Model Accuracy: {:.2f}%".format(model_accuracy * 100))

# Generating Classification report for model's performance in each class
cl_report = classification_report(y_test, predictions)
print(cl_report)
```

	precision	recall	f1-score	support
0	0.69	0.83	0.76	150
1	0.79	0.63	0.70	150
accuracy			0.73	300
macro avg	0.74	0.73	0.73	300
weighted avg	0.74	0.73	0.73	300

Gambar 4.22 Klasifikasi Kinerja Model *Epoch* 10

2) *Epoch* 20

Hasil dari 20 *epoch* dapat dilihat pada gambar 4.23.

```
model_accuracy = accuracy_score(y_test, predictions)
print("Model Accuracy: {:.2f}%".format(model_accuracy * 100))

# Generating Classification report for model's performance in each class
cl_report = classification_report(y_test, predictions)
print(cl_report)
```

	precision	recall	f1-score	support
0	0.67	0.87	0.76	150
1	0.81	0.58	0.68	150
accuracy			0.72	300
macro avg	0.74	0.72	0.72	300
weighted avg	0.74	0.72	0.72	300

Gambar 4.23 Klasifikasi Kinerja Model *Epoch* 20

3) *Epoch 30*

Hasil dari 30 *epoch* dapat dilihat pada gambar 4.23.

```
model_accuracy = accuracy_score(y_test, predictions)
print("Model Accuracy: {:.2f}%".format(model_accuracy * 100))

✓ 0.0s
Model Accuracy: 72.33%

# Generating Classification report for model's performance in each class
cl_report = classification_report(y_test, predictions)
print(cl_report)

✓ 0.0s
```

	precision	recall	f1-score	support
0	0.67	0.87	0.76	150
1	0.82	0.57	0.67	150
accuracy			0.72	300
macro avg	0.75	0.72	0.72	300
weighted avg	0.75	0.72	0.72	300

Gambar 4.24 Klasifikasi Kinerja Model *Epoch 30*

4) *Epoch 40*

Hasil dari 40 *epoch* dapat dilihat pada gambar 4.25.

```
model_accuracy = accuracy_score(y_test, predictions)
print("Model Accuracy: {:.2f}%".format(model_accuracy * 100))

Model Accuracy: 71.67%

# Generating Classification report for model's performance in each class
cl_report = classification_report(y_test, predictions)
print(cl_report)
```

	precision	recall	f1-score	support
0	0.68	0.81	0.74	150
1	0.76	0.63	0.69	150
accuracy			0.72	300
macro avg	0.72	0.72	0.71	300
weighted avg	0.72	0.72	0.71	300

Gambar 4.25 Klasifikasi Kinerja Model *Epoch 40*

Ini bertujuan untuk memberikan analisis detail tentang performa model pada setiap kelas dan mengidentifikasi kelemahan mode serta Perbaikan Model.

4.1.13 Confusion Matrix

Tahap berikutnya menghasilkan *Confusion Matrix* dari hasil prediksi suatu model klasifikasi. *Confusion Matrix* adalah *matriks* yang digunakan untuk mengevaluasi kinerja model klasifikasi dengan membandingkan prediksi model terhadap label sebenarnya.

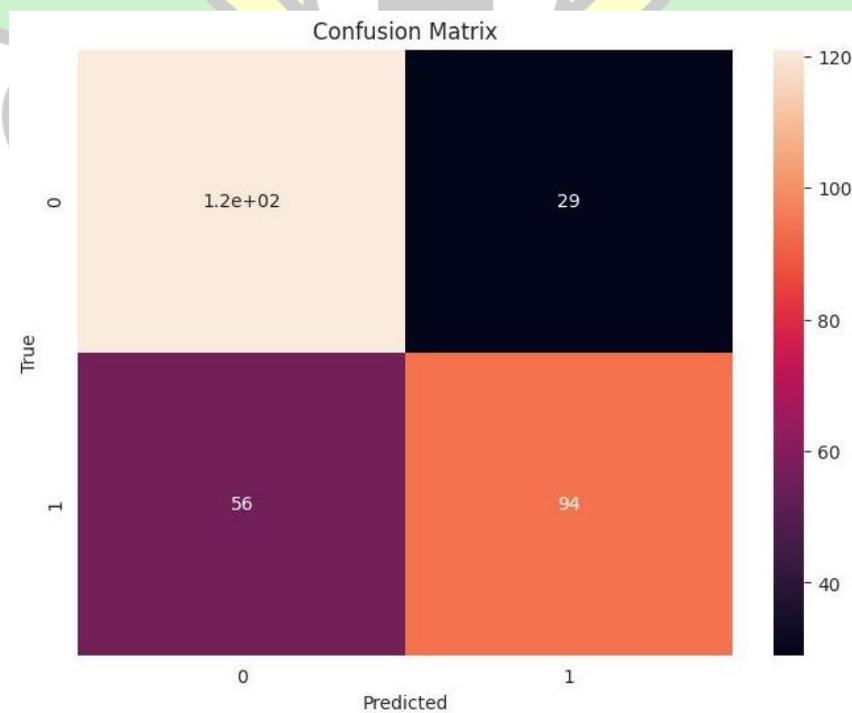
```
[31] # Menghasilkan Matriks Kebingungan untuk prediksi terhadap label sebenarnya
      cn_matrix= confusion_matrix(y_test, predictions)
      cn_matrix
Python
```

Gambar 4.26 Memanggil *Confusion Matrix*

Berikutnya adalah visualisasi *confusion matrix* menggunakan pustaka *seaborn* dan *matplotlib*.

```
# Plotting the True Positives, True Negatives, False Positives and False Negatives from model's predictions
f, ax = plt.subplots(figsize = (8,6))
ax = sns.heatmap(cn_matrix, annot=True)
ax.set_xlabel("Predicted")
ax.set_ylabel("True")
ax.set_title("Confusion Matrix")
```

Gambar 4.27 Visualisasi *Confusion Matrix*



Gambar 4.28 Hasil *Confusion Matrix*

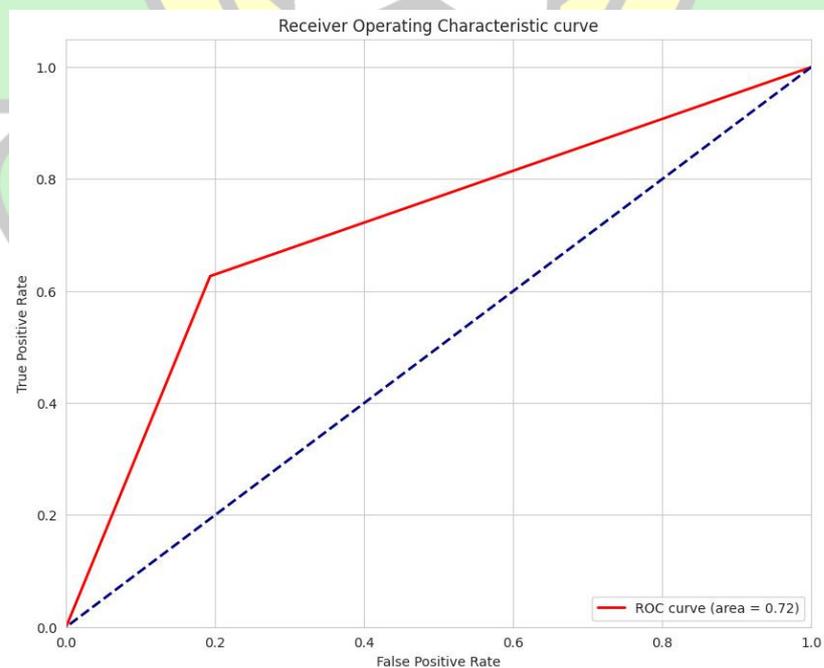
4.1.14 Kurva AOC – ROC

Tahap berikutnya adalah membuat kurva ROC (*Receiver Operating Characteristic*) untuk mengevaluasi kinerja model klasifikasi. Kurva ROC digunakan untuk mengevaluasi kemampuan model dalam membedakan antara kelas positif dan negatif. Nilai AUC (*Area Under Curve*) yang mendekati 1 menunjukkan kinerja model yang baik.

```
# Let's plot the AUC ROC curve to assess the performance of our model
fpr, tpr, _ = roc_curve(y_test, predictions)
roc_auc = auc(fpr, tpr)

plt.figure(figsize = (10,8))
plt.plot(fpr, tpr, color = 'red', lw = 2, label = 'ROC curve (area = %0.2f)' % roc_auc)
plt.plot([0, 1], [0, 1], color = 'navy', lw = 2, linestyle = '--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver Operating Characteristic curve')
plt.legend(loc = 'lower right')
```

Gambar 4.29 Memanggil Kurva AUC – ROC



Gambar 4.30 Grafik Kurva AUC - ROC

4.1.15 Hasil Prediksi

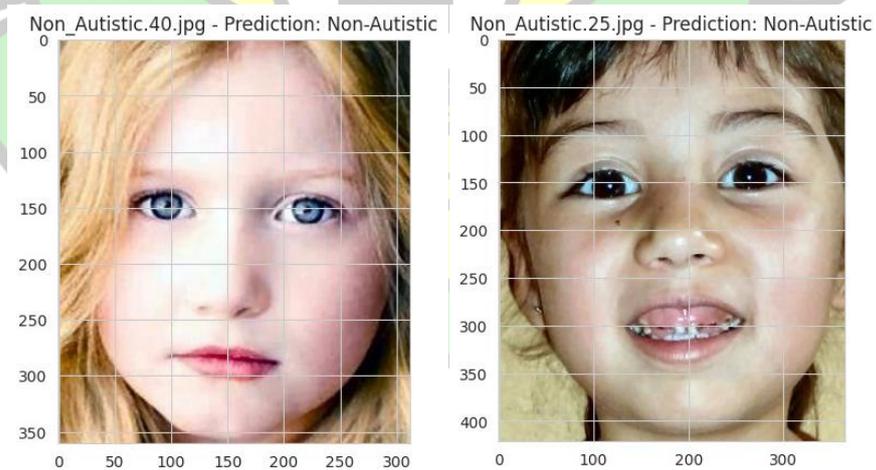
Hasil dari penelitian ini adalah menampilkan gambar uji dari *dataset* dan menampilkan prediksi model klasifikasi terhadap gambar tersebut, baik "*Autistic*" atau "*Non-Autistic*", dapat dilihat pada gambar 4.25.

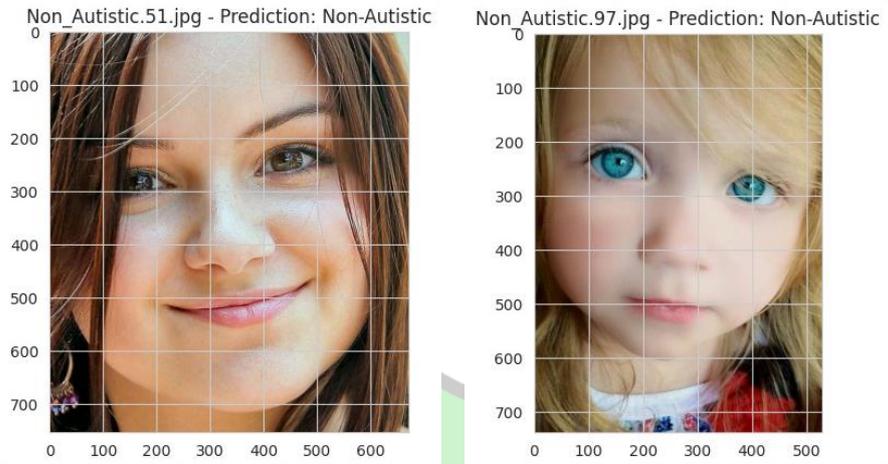
```
# Let's check our predictions against some test images
plt.figure(figsize=(4,4))
for val, i in enumerate(test_imgs[:10]):
    img = mpimg.imread(i)
    imgplot = plt.imshow(img)
    plt.title(os.path.basename(i) + ' - Prediction: ' + f"{'Autistic' if predictions[val] == 1 else 'Non-Autistic'}")
    plt.show()
```

Gambar 4.31 Cek Hasil Prediksi

Tujuan dari *code* ini adalah menampilkan gambar dari *dataset* uji beserta hasil prediksi model untuk masing-masing gambar, memberikan konteks visual tentang performa model klasifikasi yang digunakan.

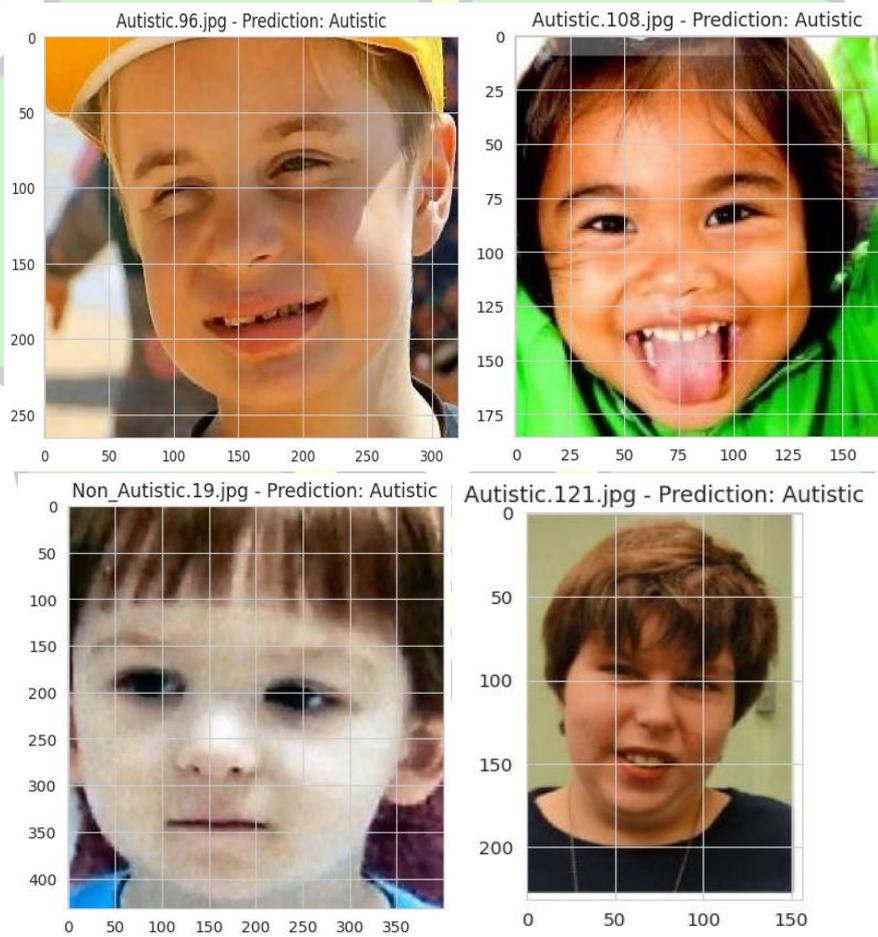
Berikut adalah hasil dari model klasifikasi gambar yang digunakan untuk memprediksi apakah seseorang dalam gambar termasuk kategori "*Autistic*" atau "*Non-Autistic*" berdasarkan ciri-ciri visual.





Gambar 4.32 Hasil Prediksi *Non-Austism*

Berikut adalah hasil uji gambar Austis dapat dilihat pada gambar 4.27.



Gambar 4.33 Hasil Prediksi *Austism*

4.1.16 Label Prediksi Gambar

Hasil dari prediksi gambar dibentuk dalam tabel. Tabel dapat dilihat pada tabel 4.1

Tabel 4.1 Label Prediksi Gambar

No	Filename	Test table	Prediction
0	<i>Non_Austictic.15</i>	0	1
1	<i>Autistic.52</i>	1	1
2	<i>Autistic.51</i>	1	1
3	<i>Non_Austictic.54</i>	0	1
4	<i>Non_Austictic.199</i>	0	0
...
295	<i>Autistic.127</i>	1	0
296	<i>Non_Austictic.28</i>	0	0
297	<i>Autistic.97</i>	1	1
298	<i>Non_Austictic.143</i>	0	0
299	<i>Non_Austictic.86</i>	0	0

Penjelasan tabel 4.1 merupakan hasil dari data test awal , 0 untuk gambar yang *non_autistic* dan 1 untuk data gambar yang autis, dan hasil yang sudah diprediksi memakai vgg16 , dapat dilihat dari table 4.1 hasil data test awal berubah ketika sistem sudah dijalankan.

4.1.17 Hasil Akurasi

Hasil akurasi dirangkup dalam bentuk tabel. Tabel hasil akurasi dari *confusion matrix* yang telah di uji dapat dilihat pada tabel 4.2.

Tabel 4.2 *Confusion Matrix*

TABEL *CONFUSION MATRIX* 300 data *testing epoch* 40

Prediction	1	0
Actual		
0	121	29
1	56	94

True Positive = 121

False Positive = 29

True Negative = 94

False Negative = 56

$$\begin{aligned} \text{Accuracy} &= \frac{TP + TN}{TP + TN + FP + FN} \\ &= \frac{121 + 94}{121 + 94 + 29 + 56} = \frac{215}{300} = 0,71 \\ 0,71 \times 100 &= 71\% \end{aligned}$$

$$\begin{aligned} \text{Sensitivity} &= \frac{TP}{TP + FN} \\ &= \frac{121}{121 + 56} = \frac{121}{177} = 0,68 \\ 0,68 \times 100 &= 68\% \end{aligned}$$

$$\begin{aligned} \text{Precision} &= \frac{TP}{TP + FP} \\ &= \frac{121}{121 + 29} = \frac{121}{150} = 0,80 \\ 0,80 \times 100 &= 80\% \end{aligned}$$

$$\begin{aligned} \text{F1 - Score} &= 2 \times \frac{\text{Precision} \times \text{Sensitivity}}{\text{Precision} + \text{Sensitivity}} \\ &= 2 \times \frac{0,80 \times 0,68}{0,80 + 0,68} = 2 \times \frac{0,544}{1,48} = 0,73 \\ 0,73 \times 100 &= 73\% \end{aligned}$$

Tabel nilai yang telah di uji dapat dilihat pada tabel 4.3.

Tabel 4.3 Nilai Akurasi

<i>Target</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>	<i>Support</i>
0	0.68	0.81	0.74	150
1	0.76	0.63	0.69	150
<i>Accuracy</i>	0.71	0.71	0.72	300
<i>Macro avg</i>	0.72	0.72	0.71	300
<i>Weighted avg</i>	0.72	0.72	0.71	300

Dari tabel 4.2 dan 4.3 serta perhitungan yang dilakukan , maka dapat dilihat pada tabel 4.4:

Tabel 4.4 Tabel Kesimpulan

Data Testing	300 data testing	0	1
<i>Accuracy</i>	0,71%	--	--
<i>Precision</i>	0,80%	0,68	0,76
<i>Recall</i>	0,68%	0,81	0,63
<i>F1-Score</i>	0,73%	0,74	0,69

Hasil yang sudah didapatkan dari penelitian ini dengan data testing sebanyak 300 data gambar mendapatkan nilai dari *accuracy*, *precision*, *recall*, *f1-score* dan dapat disimpulkan akurasi yang diperoleh sebesar 71%.

4.2 Pembahasan Hasil Penelitian

Berdasarkan hasil penelitian ini yaitu “Pengukuran Tingkat Akurasi Deteksi Wajah Autis Menggunakan CNN”. Penelitian ini menggunakan VGG16 Data *testing* yang digunakan untuk penelitian ini terdiri dari 1270 data *autism* dan 1270 *non autism*, data *testing* yang diambil dari data *training* yaitu sebanyak 300 gambar yang diambil secara acak, pada penelitian ini sendiri melakukan beberapa kali *training* model *epoch* 10, 20, 30, dan 40. Dari hasil percobaan dari keempat nilai

epoch tersebut, maka di dapatkan *epoch* tertinggi yaitu 40 *epoch* yang nilai akurasi 71%.

Pada penelitian sebelumnya Irfan Maulana, Nabila Khairunnisa, Ratna Mufidah 2023 penelitian mereka menyakut tentang “Deteksi Bentuk Wajah Menggunakan *Convolutional Neural Network* (CNN)”. Penelitian ini juga melakukan percobaan dari 10 *epoch* hingga 40 *epoch*. Namun, penelitian ini menggunakan data *training* sebanyak 800 data, yang dimana hasil akurasi tertingginya adalah 74%.

Sedangkan pada penelitian yang dilakukan oleh Israr Ahmad dkk, 2024 yang dimana mereka meneliti tentang “Deteksi Gangguan Spektrum Autisme Menggunakan Citra Wajah” yang dimana penelitian ini juga melakukan percobaan beberapa *epoch* dan menggunakan 2940 data yang masing-masing berjumlah 1470 *autism* dan 1470 *non-autism* yang dimana akurasi model VGG16 adalah 80% dan VGG19 86%. Berbeda dengan penelitian yang peneliti lakukan yang dimana hanya menggunakan satu model yaitu VGG16 dengan tingkat akurasi 71%.

Pada penelitian Deva Ega Marinda, Imam Husni Al Amin, 2023 mereka meneliti tentang “Implementasi Metode *Convolutional Neural Network* untuk Deteksi Penggunaan Masker secara *Real-Time*”. Pada penelitian menetapkan 50 *epoch* tanpa melakukan percobaan nilai *epoch* lain dan *dataset* yang digunakan sebanyak 1.300 data, yang dimana hasil akurasi mendapatkan skor 97,5%.

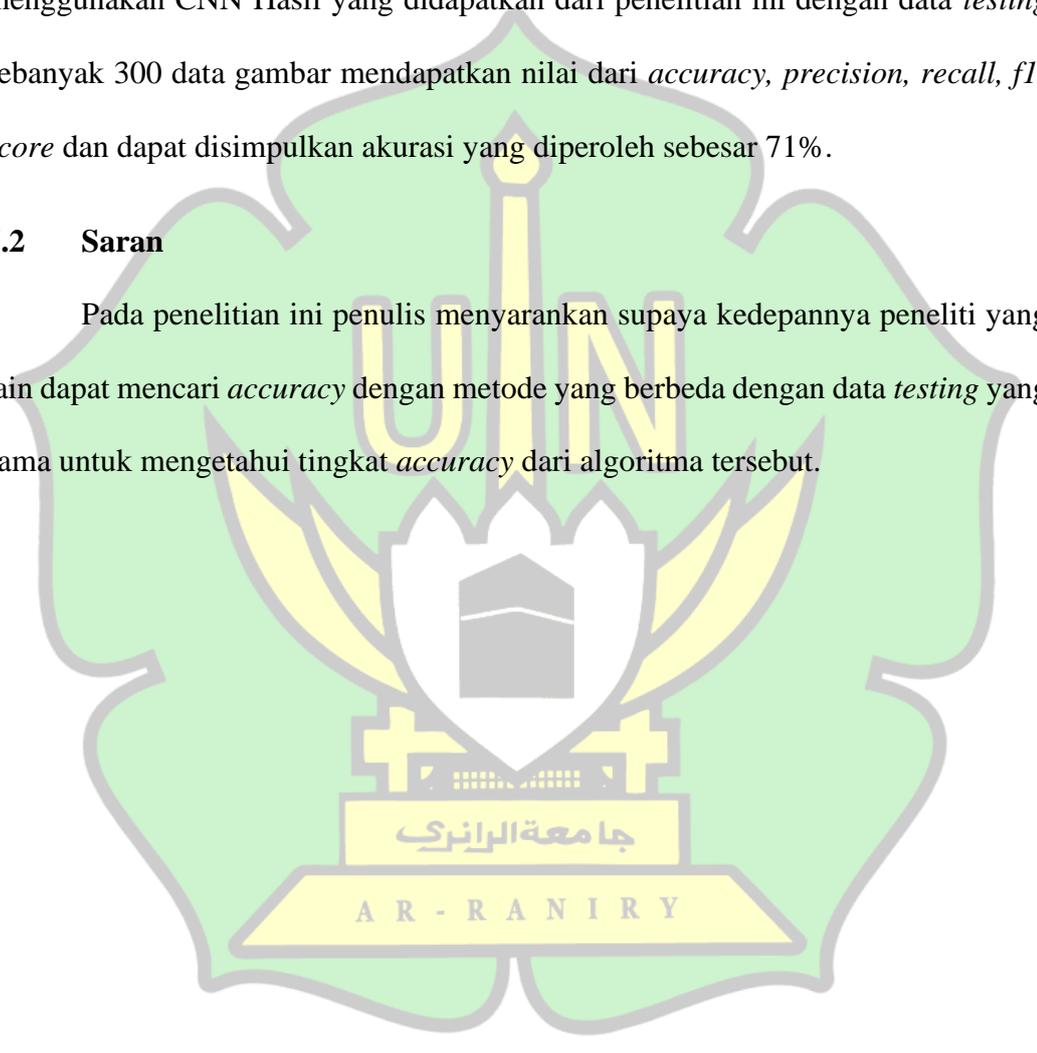
BAB V KESIMPULAN DAN SARAN

5.1 Kesimpulan

Setelah dilakukan penelitian ini hasil yang didapatkan dengan menggunakan CNN Hasil yang didapatkan dari penelitian ini dengan data *testing* sebanyak 300 data gambar mendapatkan nilai dari *accuracy*, *precision*, *recall*, *f1-score* dan dapat disimpulkan akurasi yang diperoleh sebesar 71%.

5.2 Saran

Pada penelitian ini penulis menyarankan supaya kedepannya peneliti yang lain dapat mencari *accuracy* dengan metode yang berbeda dengan data *testing* yang sama untuk mengetahui tingkat *accuracy* dari algoritma tersebut.



DAFTAR PUSTAKA

- Ameen, A., & Kar, A. K. (2021). Cross-ethnic facial recognition and autism detection using deep learning models. *Journal of Computer Vision and Image Understanding*, 207, 103194.
- Brynjolfsson, E., & McElheran, K. (2016). *The Second Machine Age: Work, Progress, and Prosperity in a Time of Brilliant Technologies*. W. W. Norton & Company.
- Chicco, D., & Jurman, G. (2020). The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation. *BMC Genomics*, 21, 6.
- Cohn, J. F., & De la Torre, F. (2015). *The Facial Action Coding System (FACS) and Emotion Recognition*. In *Handbook of Face Recognition* (pp. 469-491).
- Gosselin, P., & Schyns, P. G. (2020). Facial expression analysis in autism spectrum disorder using convolutional neural networks. *IEEE Transactions on Affective Computing*, 11(1), 98-111.
- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Kasyanov, I., & Kasyanov, I. (2020). *Facial Expression Recognition for Autism Spectrum Disorder: A Review of the Literature*. *IEEE Access*, 8, 23814-23828.
- Khan, M. A., & Khan, S. U. (2022). Deep learning approaches for autism detection: a comprehensive review. *Journal of Biomedical Informatics*, 125, 103948.
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). *ImageNet Classification with Deep Convolutional Neural Networks*. In *Advances in Neural Information Processing Systems* (pp. 1097-1105).
- Kumar, V., & Patel, P. (2021). Evaluating the accuracy of autism detection through facial expression recognition using deep learning. *International Journal of Computer Applications*, 175(5), 12-19.

- LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436-444.
- Liu, W., Li, M., & Yi, L. (2016). Identifying children with autism spectrum disorder based on their face processing abnormality: A machine learning framework. *Autism Research*, 9(8): 888-898.
- Masi, I., Chen, J., & Chang, S.-F. (2022). The latest advances in face detection: a review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(3), 1021-1038.
- Nagarajan, R., & Sinha, P. (2023). Enhancing facial recognition systems for autism spectrum disorder using advanced deep learning techniques. *Journal of Artificial Intelligence Research*, 72, 211-227.
- Pan, S. J., & Yang, Q. (2023). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10), 1345-1359.
- Szeliski, R. (2022). *Computer Vision: Algorithms and Applications* (2nd ed.). Springer. doi:10.1007/978-3-030-30432-8
- Tamilarasi, F. C., & Shanmugam, J. (2020). Convolutional Neural Network based Autism Classification. In 2020 5th International Conference on Communication and Electronics Systems (ICCES). IEEE.
- Viola, P., & Jones, M. J. (2001). Rapid object detection using a boosted cascade of simple features. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 511-518. doi:10.1109/CVPR.2001.99051