

**IDENTIFIKASI PENYAKIT PADA CITRA *CT SCAN* MENGGUNAKAN  
METODE *CONVOLUTIONAL NEURAL NETWORK***

**SKRIPSI**

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



**10584111320**

**PROGRAM STUDI INFORMATIKA  
FAKULTAS TEKNIK  
UNIVERSITAS MUHAMMADIYAH MAKASSAR  
2025**



**PENGESAHAN**

Skripsi atas nama Agung Fатurohman dengan nomor induk Mahasiswa 105 84 11113 20, dinyatakan diterima dan disahkan oleh Panitia Ujian Tugas Akhir/Skripsi sesuai dengan Surat Keputusan Dekan Fakultas Teknik Universitas Muhammadiyah Makassar Nomor : 0004/SK-Y/55202/091004/2025, sebagai salah satu syarat guna memperoleh gelar Sarjana Komputer pada Program Studi Informatika Fakultas Teknik Universitas Muhammadiyah Makassar pada hari Sabtu, 30 Agustus 2025.

Panitia Ujian :

1. Pengawas Umum

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, S.T., M.T., ASEAN, Eng. :

6 Rabi'ul Awal 1447 H

30 Agustus 2025 M

2. Penguji

a. Ketua : Prof. Dr. Ir. Hafsa Nirwana, M.T.

b. Sekertaris : Titin Wahyuni, S.Pd., M.T.

3. Anggota

: Chyquitha Danuputri, S.Kom., M.Kom.

2. Muhyiddin A.M.Hayat, S.Kom., M.T.

3. Lukman, S.Kom., M.T.

Mengetahui :

Pembimbing I

Pembimbing II

Desi Anggreani, S.Kom., M.T.

Rizki Yusliana Bakti, S.T., M.T.

Dekan

Muhammad Suryadi S. Kuba, S.T., M.T.

NIM : 795 288



MAJELIS PENDIDIKAN TINGGI PIMPINAN PUSAT MUHAMMADIYAH

UNIVERSITAS MUHAMMADIYAH MAKASSAR

**FAKULTAS TEKNIK**



## 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 : IDENTIFIKASI PENYAKIT PADA CITRA CT SCAN MENGGUNAKAN METODE CONVOLUTIONAL NEURAL NETWORK

Nama : Agung Faturohman

Stambuk : 105 84 11113 20

Makassar, 8 September 2025

Telah Diperiksa dan Disetujui  
Oleh Dosen Pembimbing;

Pembimbing I

Desi Anggreani S.Kom.,M.T.

Pembimbing II

Rizki Yusliana Bakti, S.T., M.T.

Mengetahui,

Ketua Prodi Informatika



Rizki Yusliana Bakti, S.T., M.T.  
NBM : 1307 284



Kampus  
Merdeka



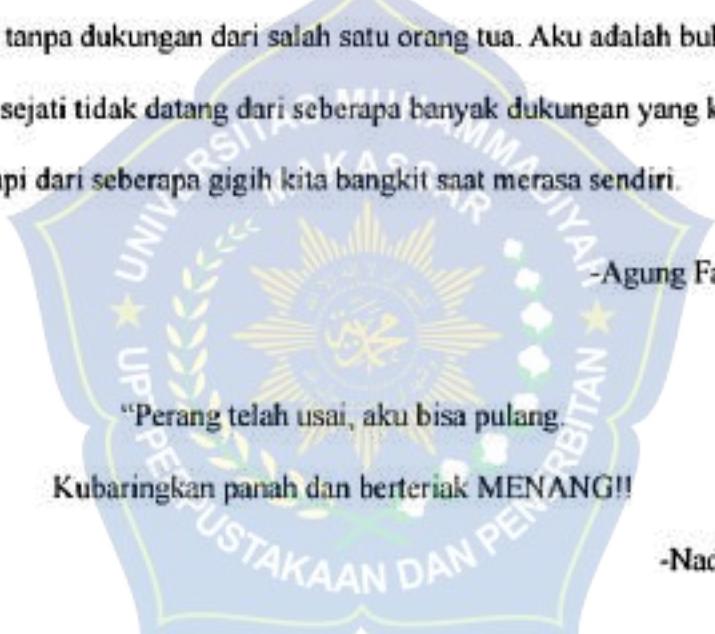
## HALAMAN MOTTO

“Tidak ada mimpi yang gagal, yang ada hanyalah mimpi yang tertunda, sekiranya  
merasa gagal dalam mencapainya, jangan khawatir, mimpi mimpi lain bisa  
 diciptakan”

-Winda basudara-

Gelar ini bukan hanya selembaran kertas, melainkan cerminan dari badai yang telah  
aku taklukkan seorang diri. Aku bangga pada diriku, karena aku tahu betapa beratnya  
perjuangan ini tanpa dukungan dari salah satu orang tua. Aku adalah bukti hidup  
bahwa kekuatan sejati tidak datang dari seberapa banyak dukungan yang kita terima,  
tetapi dari seberapa gigih kita bangkit saat merasa sendiri.

-Agung Faturohman-



“Perang telah usai, aku bisa pulang.  
Kubaringkan panah dan berteriak MENANG!!

-Nadin amizah-

## ABSTRAK

**AGUNG FATUROHMAN**, *Identifikasi Penyakit Pada Citra Ct Scan Menggunakan Metode Convolutional Neural Network* (Dibimbing oleh Desi Anggreani dan Rizki Yusliana Bakti).

Stroke iskemik merupakan salah satu penyebab utama kematian dan kecacatan, terutama pada kelompok lanjut usia. Proses diagnosis penyakit ini umumnya dilakukan melalui pemeriksaan citra CT scan kepala oleh tenaga medis, yang membutuhkan waktu dan keahlian khusus. Penelitian ini bertujuan untuk membangun model deteksi stroke iskemik menggunakan metode *Convolutional Neural Network* (CNN) dengan arsitektur MobileNetV2.

Data yang digunakan terdiri dari citra CT scan kepala pasien normal dan pasien stroke iskemik, yang telah melalui tahap pra-pemrosesan berupa resizing, augmentasi, dan normalisasi. Model CNN dibangun dengan memanfaatkan MobileNetV2 sebagai *feature extractor*, kemudian ditambahkan layer klasifikasi dan dilatih menggunakan data tersebut. Evaluasi dilakukan untuk mengukur performa model dalam mengenali kondisi otak normal dan yang mengalami stroke. Hasil pengujian menunjukkan bahwa model memiliki akurasi sebesar 91,6%, dengan nilai precision, recall, dan F1-score yang tinggi. Hal ini menunjukkan bahwa model CNN yang dibangun mampu melakukan klasifikasi secara otomatis dengan cukup baik.

Kata kunci: Stroke Iskemik, *CT Scan*, CNN, MobileNetV2, Klasifikasi Citra, *Deep Learning*.

## ABSTRACT

**AGUNG FATUROHMAN**, Disease Identification on CT Scan Images Using Convolutional Neural Network (Supervised by Desi Anggreani san Rizki Yusliana Bakti)

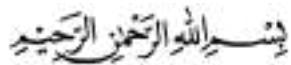
Ischemic stroke is one of the leading causes of death and disability, particularly among the elderly. The diagnosis process is generally carried out through the analysis of brain CT scan images by medical professionals, which requires time and specific expertise. This study aims to develop an ischemic stroke detection model using the Convolutional Neural Network (CNN) method with the MobileNetV2 architecture.

The data used consists of head CT scan images of both normal patients and patients with ischemic stroke. The data underwent preprocessing steps including resizing, augmentation, and normalization. The CNN model was built using MobileNetV2 as a feature extractor, followed by additional classification layers and trained on the preprocessed dataset. The model was then evaluated to measure its performance in recognizing normal and stroke-affected brain images.

The test results show that the model achieved an accuracy of 91.6%, along with high values in precision, recall, and F1-score. These results indicate that the CNN-based model is capable of automatically classifying CT scan images with good accuracy. This model has the potential to be used as an early diagnostic tool for ischemic stroke in clinical settings.

Keywords: *Ischemic Stroke, CT Scan, CNN, MobileNetV2, Image Classification, Deep Learning.*

## KATA PENGANTAR



*Assalamu'alaikum warahmatullahi wabarakatuh,*

*Alhamdulillahi rabbil 'alamin*, senantiasa penulis panjatkan rasa syukur kepada Sang Pencipta, Allah SWT, atas limpahan rahmat, ridha, hidayah, dan karunia-Nya, sehingga penulis dapat menyelesaikan skripsi dengan judul **“Identifikasi Penyakit Pada Citra CT Scan Menggunakan Metode Convolutional Neural Network”**. Shalawat dan salam penulis panjatkan kepada Nabi Besar Muhammad SAW, sebagai *uswatun hasanah dan rahmatan lil- 'alamin*, yang telah membawa umat manusia dari zaman jahiliah menuju era modern seperti yang kita rasakan saat ini.

Selama proses penyusunan skripsi ini, penulis menerima banyak bimbingan, arahan, motivasi, dan bantuan dari berbagai pihak, baik secara langsung maupun tidak langsung. Oleh karena itu, penulis ingin menyampaikan rasa hormat dan terima kasih kepada:

1. Bapak Dr. Ir. H. Abd Rakhim Nanda, S.T., M.T IPU. sebagai Rektor Universitas Muhammadiyah Makassar.
2. Bapak Ir. Muhammad Syafa'at S.Kuba. ST., M.T Sebagai Dekan Fakultas Teknik Universitas Muhammadiyah Makassar.
3. Ibu Rizki Yusliana Bakti, S.T., M.T sebagai Ketua Prodi Informatika Fakultas Teknik Universitas Muhammadiyah Makassar.
4. Ibu Desi Anggreani, S.Kom., M.T sebagai pembimbing I yang dengan telah Ikhlas memberikan bimbingan dan arahan selama penyusunan tugas akhir ini.
5. Ibu Rizki Yusliana Bakti, S.T., M.T sebagai pembimbing II yang dengan telah Ikhlas memberikan bimbingan dan arahan selama penyusunan vi tugas akhir ini.
6. Segenap Bapak – bapak dan Ibu Dosen Prodi Informatika Fakultas Teknik Universitas Muhammadiyah Makassar yang telah memberikan bakat dan ilmu pengetahuan serta mendidik penulis selama proses belajar mengajar di Universitas Muhammadiyah Makassar.
7. Bapak/Ibu Pegawai RSUD Labuang Baji, yang telah memberikan bantuan, izin pengumpulan data, serta informasi yang sangat berarti dalam kelancaran penelitian ini.

8. Kedua orang tua yang tercinta, yaitu Adam dan Roswati. Penulis mengucapkan terima kasih yang sebesar-besarnya atas segala doa, kasih sayang dan dukungan baik secara moral maupun materi.
9. Rekan-rekan mahasiswa utamanya angkatan 2020 Fakultas Teknik Universitas Muhammadiyah Makassar terima kasih atas dukungan dan kerjasamanya selama menempuh Pendidikan serta penyelesaian penyusunan proposal skripsi ini.
10. Senior dan teman dalam grup balanda yang selalu bersama, penulis mengucapkan terima kasih yang sebesar-besarnya atas bantuan dan dukungannya baik secara moral maupun materi.
11. Terakhir, terima kasih untuk diri sendiri, karena telah mampu berusaha keras dan berjuang sampai sejauh ini meskipun menghadapi berbagai masalah, termasuk dalam keluarga

Semoga semua pihak tersebut di atas mendapat pahala yang berlipat ganda disisi Allah SWT dan Tugas Akhir yang sederhana ini dapat bermanfaat bagi penulis, rekan – rekan, Masyarakat serta Bangsa dan Negara.

Aamiin Billahi Fisabilhaq, Fastabiqul Khairat.

Wassalamualaikum Wr. Wb.

Makassar, 11 Agustus 2025

Agung Faturohman

## DAFTAR ISI

DAFTAR ISI.....	vi
BAB I.....	1
PENDAHULUAN.....	1
A. Latar Belakang.....	1
B. Rumusan Masalah.....	3
C. Tujuan Penelitian .....	3
D. Manfaat Penelitian .....	3
E. Ruang Lingkup Penelitian.....	5
F. Sistematika Penulisan .....	5
BAB II TINJAUAN PUSTAKA.....	7
A. Landasan Teori.....	7
B. Penelitian Terkait.....	21
C. Kerangka Pikir .....	24
BAB III.....	25
METODE NELITIAN .....	25
A. Tempat dan Waktu Penelitian .....	25
B. Alat dan Bahan .....	25
C. Perancangan Sistem .....	25
D. Teknik Pengujian Sistem .....	31
E. Teknik Analisis Data.....	34
BAB IV HASIL DAN PEMBAHASAN.....	37
A. Hasil dan Gambaran Umum Data .....	37
B. Tahap Proprecessing.....	39
C. Implementasi Model CNN.....	41
1. Arsitektur Model CNN .....	41
2. Analisis Implemetasi Model CNN.....	42
3. Visualisasi Loss dan Akurasi Model CNN .....	44
D. Hasil Pengujian Model .....	46
a. Evaluasi Model Confusion Matrix .....	47
b. Analisi Kesalahan .....	51

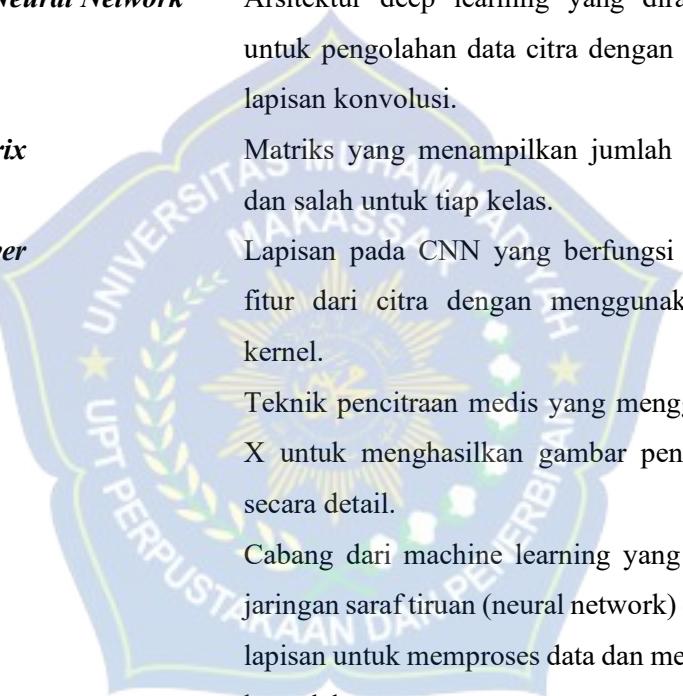
<b>BAB V</b> .....	<b>53</b>
<b>PENUTUP</b> .....	<b>53</b>
A. Kesimpulan.....	53
B. Saran.....	53
<b>DAFTAR PUSTAKA</b> .....	<b>54</b>



## DAFTAR GAMBAR

Gambar 2. 1 Normal dan Stroke Iskemik.....	9
Gambar 2. 2 Citra Thresholding (Situngkir, 2019) .....	13
Gambar 2. 3 Arsitektur Convolutional Neural Network (Situngkir, 2019) .....	16
Gambar 2. 4 Fully Connected Layer (Situngkir, 2019). .....	20
Gambar 2. 5 Kerangka Pikir.....	24
Gambar 3. 1 Flowchart Perancangan Sistem .....	26
Gambar 4. 1Dataset CT Scan Normal.....	37
Gambar 4. 2 Dataset CT Scan Sakit Stroke Iskemik .....	38
Gambar 4. 3 Tahap Proprecessing Resizing .....	39
Gambar 4. 4 Identifikasi Sakit Stroke Iskemik Dan Normal .....	43
Gambar 4. 5 Grafik Training dan loss.....	45
Gambar 4. 6 Performa Hasil Evaluasi Model .....	47
Gambar 4. 7 Confusion Matrix .....	50
Gambar 4. 8 Analisis Kesalahan False Positive.....	51
Gambar 4. 9 Analisis Kesalahan False Negatif.....	52

## DAFTAR ISTILAH



<b>Accuracy</b>	Percentase prediksi model yang benar dari seluruh data yang diuji.
<b>Augmentasi Data</b>	Proses memperbanyak variasi data dengan transformasi seperti rotasi, flipping, atau zooming untuk mencegah overfitting.
<b>Batch Size</b>	Jumlah data yang diproses sekaligus dalam satu langkah pelatihan.
<b>Convolutional Neural Network</b>	Arsitektur deep learning yang dirancang khusus untuk pengolahan data citra dengan memanfaatkan lapisan konvolusi.
<b>Confusion Matrix</b>	Matriks yang menampilkan jumlah prediksi benar dan salah untuk tiap kelas.
<b>Convolution layer</b>	Lapisan pada CNN yang berfungsi mengekstraksi fitur dari citra dengan menggunakan filter atau kernel.
<b>CT Scan</b>	Teknik pencitraan medis yang menggunakan sinar-X untuk menghasilkan gambar penampang tubuh secara detail.
<b>Deep Learning</b>	Cabang dari machine learning yang menggunakan jaringan saraf tiruan (neural network) dengan banyak lapisan untuk memproses data dan mengekstrak pola kompleks.
<b>Epoch</b>	Satu kali siklus pelatihan model menggunakan seluruh data pelatihan.
<b>F1 Score</b>	Rata-rata harmonis antara precision dan recall.
<b>Flip</b>	Proses membalik citra secara horizontal atau vertikal untuk menambah variasi data pelatihan.
<b>Full Connected Layer</b>	Lapisan pada jaringan saraf di mana setiap neuron terhubung ke semua neuron pada lapisan sebelumnya, biasanya digunakan di akhir model untuk menghasilkan output akhir.

<b>Loss Function</b>	Fungsi yang mengukur selisih antara prediksi model dan label sebenarnya, misalnya categorical crossentropy.
<b>MobileNetV2</b>	Model <i>pre-trained</i> ringan dan efisien untuk klasifikasi citra, dirancang untuk perangkat dengan sumber daya terbatas, dengan input standar $224 \times 224$ piksel.
<b>Normalisasi</b>	Mengubah nilai piksel citra ke skala tertentu (misalnya 0–1 atau -1–1) agar memudahkan proses pelatihan.
<b>Optimizer</b>	Algoritma yang mengatur pembaruan bobot model selama pelatihan, misalnya Adam.
<b>Output</b>	Hasil akhir yang diberikan oleh model setelah proses prediksi selesai, biasanya berupa probabilitas atau label kelas.
<b>Overtfitting</b>	Kondisi saat model terlalu menyesuaikan diri dengan data pelatihan sehingga performanya buruk pada data baru.
<b>Padding</b>	Teknik menambahkan piksel di tepi citra sebelum proses konvolusi untuk mempertahankan ukuran output atau mengontrol dimensi hasil.
<b>Pooling Layer</b>	Lapisan pada CNN yang berfungsi untuk mengurangi dimensi citra dan jumlah parameter, misalnya dengan metode max pooling atau average pooling.
<b>Precision</b>	Proporsi prediksi positif yang benar dibandingkan seluruh prediksi positif.
<b>Preprocessing</b>	Tahap awal pemrosesan data citra sebelum masuk ke model, meliputi resizing, normalisasi, dan augmentasi
<b>Python</b>	Bahasa pemrograman tingkat tinggi yang populer digunakan untuk pengembangan aplikasi, termasuk pembelajaran mesin dan deep learning.

<b>Recall</b>	Proporsi data positif yang benar-benar terdeteksi dibandingkan seluruh data positif
<b>Rescale</b>	Proses mengubah nilai piksel citra ke skala tertentu, biasanya membagi dengan angka 255 agar berada pada rentang 0–1.
<b>Resizing</b>	Mengubah ukuran citra ke dimensi yang sesuai untuk model, misalnya $224 \times 224$ piksel pada MobileNetV2.
<b>Rotation</b>	Transformasi citra dengan memutarnya pada sudut tertentu untuk memperkaya variasi data pelatihan.
<b>Shear</b>	Transformasi citra dengan menggeser sebagian piksel secara miring untuk menghasilkan distorsi terkontrol.
<b>Stride</b>	Jumlah langkah pergeseran filter atau kernel saat melakukan proses konvolusi pada citra.
<b>Stroke Iskemik</b>	Jenis stroke yang terjadi akibat penyumbatan pembuluh darah di otak sehingga aliran darah terhambat.
<b>Testing Data</b>	Data yang digunakan untuk menguji performa model setelah pelatihan selesai, biasanya belum pernah dilihat model sebelumnya.
<b>Training Data</b>	Data yang digunakan untuk melatih model agar dapat mengenali pola dan melakukan prediksi yang tepat.
<b>Zoom</b>	Proses memperbesar atau memperkecil citra untuk menambah variasi data pelatihan.

## **BAB I**

### **PENDAHULUAN**

#### **A. Latar Belakang**

Indonesia merupakan negara berkembang yang saat ini mengalami perubahan epidemiologi transisi penyakit, yaitu adanya peningkatan kejadian penyakit tidak menular. World Health Organization (WHO) memperkirakan Penyakit Tidak Menular (PTM) menjadi penyebab terbesar kematian di seluruh dunia, yaitu sekitar 40 juta orang meninggal setiap tahun disebabkan oleh PTM (71% dari semua kematian). Berdasarkan Riset Kesehatan Dasar tahun 2018 (Kemenkes RI, 2018) dan Survei Kesehatan Indonesia 2023, PTM mengalami peningkatan, salah satunya stroke (Romdzati & Firmawati, 2024).

Stroke merupakan salah satu gangguan kardiovaskuler yang mempengaruhi arteri penting yang menuju ke otak. Stroke terjadi ketika pembuluh darah yang mengangkut oksigen dan nutrisi menuju otak terblokir oleh bekuan maupun pecahan, sehingga otak tidak mendapatkan darah yang dibutuhkan, selanjutnya sel-sel otak mengalami kematian. WHO mengungkapkan bahwa stroke merupakan gangguan fungsi otak yang ditandai dengan berkembang cepatnya tanda-tanda fokal maupun global, berlangsung lebih dari 24 jam (bisa menimbulkan kematian), dengan sebab yang jelas karena vaskuler (Romdzati & Firmawati, 2024).

Stroke iskemik adalah penyakit atau gangguan fungsional otak berupa kelumpuhan saraf yang diakibatkan oleh gangguan aliran darah pada salah satu bagian otak. Gangguan saraf maupun kelumpuhan yang terjadi tergantung pada bagian otak mana yang terkena. Penyakit ini dapat sembuh sempurna, sembuh dengan cacat atau kematian (Hisni et al., 2022).

Menurut *World Health Organization*, stroke iskemik disebabkan oleh gangguan suplai darah ke otak, biasanya karena pecahnya pembuluh darah atau penyumbatan oleh gumpalan darah. Hal ini memotong pasokan oksigen dan nutrisi, menyebabkan kerusakan pada jaringan otak. Gejala yang paling umum dari stroke iskemik adalah kelemahan mendadak atau mati rasa pada wajah, lengan atau kaki, paling sering pada satu sisi tubuh. Untuk mencapai hasil fungsional yang baik, diperlukan tata laksana yang tepat dan cepat. Saat ini, stroke iskemik di Indonesia menjadi salah satu penyakit utama penyebab disabilitas dan mortalitas pada pasien lansia (Togu et al., 2021).

Penerapan deep learning pada analisis citra medis di Indonesia dapat memberikan solusi terhadap keterbatasan tersebut, khususnya dalam mendukung daerah-daerah yang mengalami keterbatasan tenaga ahli medis. Namun, masih ada tantangan besar yang harus diatasi, termasuk infrastruktur medis yang memadai, dan peraturan yang mengatur penggunaan teknologi ini (Khairi et al., 2024).

Menurut (Situngkir, 2019), CNN merupakan salah satu metode yang sering digunakan dalam sistem identifikasi. CNN adalah jenis jaringan saraf tiruan yang dirancang untuk memproses data dua dimensi. Karena kemampuannya dalam menangani jaringan dengan kedalaman tinggi serta penerapannya yang umum pada pengolahan citra, CNN diklasifikasikan sebagai bagian dari Deep Neural Network.

Berdasarkan penelitian di atas, maka penelitian ingin mengidentifikasi penyakit tumor otak, jantung pada CT Scan menggunakan metode CNN bertujuan untuk mengeksplorasi dan mengoptimalkan penggunaan teknologi deep learning dalam bidang kesehatan, khususnya dalam mendeteksi penyakit seperti tumor otak,

## **B. Rumusan Masalah**

Berdasarkan latar belakang diatas, peneliti dapat merumuskan masalah sebagai berikut:

1. Bagaimana mengimplementasikan metoder CNN untuk mengidentifikasi penyakit Stroke Iskemik pada citra CT scan?
2. Bagaimana analisis tingkat akurasi metode CNN dalam mendeteksi penyakit Stroke Iskemik pada citra CT Scan?

## **C. Tujuan Penelitian**

Berdasarkan Rumusan Masalah di atas dapat disimpulkan bahwa tujuan 2 penelitian ini adalah sebagai berikut:

1. Untuk mengimplementasikan metode CNN untuk mengidentifikasi penyakit Stroke Iskemik pada citra CT scan?
2. Untuk menganalisis tingkat akurasi metode CNN dalam mendeteksi penyakit Stroke Iskemik pada citra CT Scan?

## **D. Manfaat Penelitian**

Penelitian tentang mengedintifikasi penyakit pada CT Scan menggunakan metode convolutional neural network (CNN) diharapkan dapat bermanfaat baik secara teoritis maupun secara praktis:

1. Secara Teoritis.
  - a. Memberikan kontribusi pada pengembangan ilmu kecerdasan buatan (AI), khususnya dalam penerapan CNN untuk analisis citra medis.
  - b. Memperluas pemahaman tentang arsitektur CNN yang optimal untuk mendeteksi penyakit pada citra CT scan.

- c. Berkontribusi pada literatur akademik di bidang Computer Vision, Deep Learning, dan aplikasi teknologi AI dalam diagnosis medis.
  - d. Menyediakan wawasan mengenai teknik pengolahan data citra medis, seperti segmentasi, augmentasi, dan normalisasi, yang dapat meningkatkan performa model.
2. Secara Praktis
- a. Bagi Peneliti
    - 1) Memahami proses kerja convolutional neural network (CNN) dan bentuk identifikasi penyakit.
    - 2) Sebagai portofolio untuk peneliti yang berguna untuk masa yang akan datang.
  - b. Bagi Universitas
    - 1) Sebagai bahan referensi untuk penelitian yang akan datang.
    - 2) Sebagai bahan evaluasi bagi universitas dalam mengembangkan keilmuan, dalam hal ini yang berkaitan dengan program berbasis convolutional neural network (CNN) .
    - 3) Pengembangan Kurikulum: Memberikan materi pembelajaran dan studi kasus nyata untuk mata kuliah terkait kecerdasan buatan, pemrosesan citra, atau aplikasi teknologi dalam kesehatan.

## **E. Ruang Lingkup Penelitian**

Dari analisis rumusan masalah di atas dapat dirumuskan beberapa batasan masalah yaitu :

1. Objek penelitian identifikasi penyakit pada CR Scan di ruang radiologi rumah sakit labuang baji.
2. Metode yang digunakan adalah CNN untuk ekstraksi fitur, klasifikasi, dan deteksi penyakit pada citra CT scan.
3. Data dan Waktu menggunakan Penelitian difokuskan pada analisis citra medis berupa CT scan penyakit Stroke Iskemik. Dalam periode tertentu (misalnya 1 bulan).
4. Batasan Penelitian Fokus ini terbatas pada data CT scan yang tersedia, dengan asumsi bahwa kualitas dan anotasi data sudah sesuai standar.
5. Hasil Diharapkan dapat menghasilkan model CNN yang dapat digunakan sebagai alat bantu diagnosis dan memberikan rekomendasi untuk pengembangan lebih lanjut dalam aplikasi teknologi AI di bidang kesehatan.

## **F. Sistematika Penulisan**

Untuk memberikan gambaran umum dari seluruh penulisan ini, Adapun sistematika penulisan yaitu :

### **BAB I PENDAHULUAN**

Pada bab ini menerangkan secara singkat dan jelas mengenai latar belakang penulisan penelitian tugas akhir, rumusan masalah, tujuan dan manfaat, Batasan permasalahan, metodologi yang digunakan dan sistematika penulisan.

## BAB II TINJAUAN PUSTAKA

Pada bab ini membahas tentang teori – teori yang melandasi penulis dalam melaksanakan skripsi.

## BAB III METODE PENELITIAN

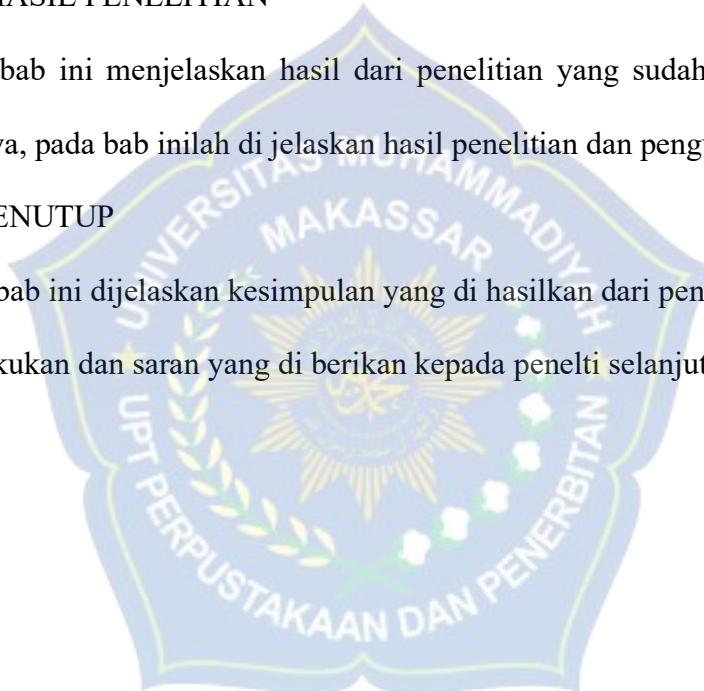
Pada bab ini membahas tentang metode penelitian dan alat yang 4 digunakan untuk pembuatan sistem.

## BAB IV HASIL PENELITIAN

Pada bab ini menjelaskan hasil dari penelitian yang sudah di lakukan sebelumnya, pada bab inilah di jelaskan hasil penelitian dan pengujian.

## BAB V PENUTUP

Pada bab ini dijelaskan kesimpulan yang di hasilkan dari penelitian yang telah di lakukan dan saran yang di berikan kepada penelti selanjutnya.



## **BAB II**

### **TINJAUAN PUSTAKA**

#### **A. Landasan Teori**

##### **1. Stroke Iskemik**

Stroke Iskemik adalah stroke yang disebabkan karena penyumbatan pembulu darah di otak sehingga oksigen ke otak berkurang dan terjadi kematian sel atau jaringan otak. Angka kejadian stroke meningkat seiring dengan bertambahnya usia, semakin tinggi usia seseorang semakin tinggi kemungkinan terjadi stroke. Faktor risiko stroke iskemik dapat dibagi menjadi dua yaitu faktor risiko yang dapat dimodifikasi dan faktor risiko yang tidak dapat dimodifikasi, mengklasifikasikan faktor resiko yang dapat dimodifikasi terdiri dari hipertensi, diabetes militus, dislipidemia, kurang aktifitas fisik, diet tidak sehat dan stress. Sedangkan faktor resiko yang tidak dapat dimodifikasi terdiri dari riwayat keluarga, umur dan jenis kelamin (Hisni et al., 2022).

Menurut penelitian (Tamburian et al., 2020). Hipertensi dapat memicu terjadinya aterosklerosis. Hal tersebut dapat mendorong Low Density Lipoprotein (LDL) kolesterol untuk lebih mudah masuk dalam lapisan intima lumen pembuluh darah dan menurunkan elastisitas dari pembuluh darah tersebut. Diabetes dapat meningkatkan risiko stroke dua kali lipat. Semakin tinggi kadar gula dalam darah, semakin mudah terkena stroke. Hiperkolesterolemia adalah peningkatan kadar kolesterol dalam darah.

Kolesterol dalam tubuh yang berlebih akan terimbun dalam dinding pembuluh darah dan dapat menimbulkan suatu kondisi yang disebut aterosklerosis yang dapat menyebabkan penyakit stroke.

Menurut World Health Organization (WHO), penyakit stroke merupakan penyakit nomor dua yang menyebabkan kematian hampir di seluruh dunia dan nomor tiga penyebab utama disabilitas. Di Amerika Serikat, stroke menjadi penyakit nomor lima yang menyebabkan kematian, setelah penyakit jantung, kanker, dan penyakit pernafasan kronis. Menurut Kementerian Kesehatan Republik Indonesia 2018, prevalensi stroke (permil) berdasarkan diagnosis pada penduduk umur lebih dari 15 tahun, Indonesia mengalami kenaikan angka kejadian stroke dari tahun 2013 sampai 2018, yaitu 2013 sebanyak 7%, sedangkan pada tahun 2018 naik menjadi 10,9%. Dengan spesifikasi laki-laki 11,0%, perempuan 10,9%. Menurut Kementerian Kesehatan Republik Indonesia 2018, prevalensi stroke (permil) berdasarkan diagnosis pada penduduk umur lebih dari 15 tahun, Indonesia mengalami kenaikan angka kejadian stroke dari tahun 2013 sampai 2018, yaitu 2013 sebanyak 7%, sedangkan pada tahun 2018 naik menjadi 10,9%. Dengan spesifikasi laki-laki 11,0%, perempuan 10,9%).

Perubahan pola hidup seperti makan tidak teratur, kurang olahraga, jam kerja berlebihan serta konsumsi makanan cepat saji sudah menjadi kebiasaan lazim yang berpotensi menimbulkan serangan stroke. Stroke merupakan masalah kesehatan dan perlu mendapat perhatian khusus. Berdasarkan kejadian yang ada dilapangan pemberian mobilisasi dini pada pasien stroke belum mendapat perhatian.

Kematian yang diakibatkan oleh stroke sekitar 10-30% pasien yang dirawat dan 70-90% pasien yang hidup pasca stroke. Sekitar 90% pasien mengalami kelumpuhan atau kelemahan separuh tubuhnya. Pasien stroke yang tidak segera mendapatkan penanganan medis dapat mengakibatkan kelumpuhan dan juga menimbulkan komplikasi, salah satunya seperti terjadinya gangguan mobilisasi, gangguan fungsional, gangguan aktivitas sehari-hari dan kecacatan yang tidak dapat disembuhkan. Gangguan fungsional yang umum terjadi pada penderita stroke ialah pada ekstremitas atas, diantaranya seperti kehilangan kontrol yang dapat menurunnya kekuatan otot (Sholihany et al., 2021).



*Gambar 2. 1 Normal dan Stroke Iskemik*

## 2. *CT Scan*

CT Scan, atau *Computed Tomography Scan*, adalah sebuah prosedur pencitraan medis yang menggunakan sinar-X dan bantuan komputer untuk menghasilkan gambar penampang tubuh secara rinci. Berbeda dengan rontgen biasa, *CT Scan* mampu menampilkan irisan melintang dari organ, tulang, pembuluh darah, maupun jaringan lunak, sehingga sangat membantu dalam

proses diagnosis berbagai kondisi medis. Alat ini bekerja dengan cara memutar sumber sinar-X di sekitar tubuh pasien dan mengambil gambar dari berbagai sudut, lalu komputer mengolah gambar-gambar tersebut menjadi visualisasi 2D atau 3D yang detail.

*CT Scan* umumnya digunakan untuk mendeteksi penyakit seperti stroke, tumor, infeksi, atau cedera dalam, serta membantu perencanaan operasi dan prosedur medis lainnya. Namun, karena melibatkan paparan radiasi, penggunaan *CT Scan* perlu diperhitungkan secara hati-hati, terutama pada anak-anak dan wanita hamil. Meskipun demikian, teknologi ini tetap menjadi salah satu alat penting dalam dunia kedokteran modern karena kemampuannya memberikan gambaran anatomi tubuh yang akurat dan cepat.

### 3. Citra digital

Citra sebagai output dari suatu sistem dapat berwujud dalam berbagai jenis sesuai dengan karakteristiknya. Misalnya, terdapat citra optik seperti foto, citra analog berupa sinyal video yang ditampilkan di televisi, dan citra digital seperti gambar serta video yang dapat disimpan secara elektronik. Bentuk citra ini muncul dalam kehidupan sehari-hari, baik sebagai citra analog maupun digital. Misalnya, citra analog dapat berupa foto yang dicetak pada kertas mengilap, lukisan yang dibuat di atas kanvas atau dinding, dan lain sebagainya.

Sifat kontinu pada citra analog terlihat dari penggunaannya pada monitor televisi, hasil X-ray, foto cetak di kertas mengilap, mural, hasil CT scan, serta gambar yang tersimpan dalam pita kaset, dan seterusnya (Situngkir, 2019).

Berbeda dengan citra analog yang tidak bisa diproses secara langsung oleh komputer, citra digital memungkinkan dilakukan komputasi karena memiliki tipe dan ukuran file tertentu. Dalam citra digital, setiap kotak kecil disebut piksel yang memiliki koordinat (x) dan (y), di mana sumbu x mewakili kolom secara horizontal dan sumbu y mewakili baris secara vertikal. Masing-masing piksel memiliki nilai yang merepresentasikan tingkat keabuan (gray level), yang menggambarkan derajat kecerahan atau kode warnanya (Situngkir, 2019).

a. *RGB Image*

Citra RGB adalah representasi ruang warna yang terdiri dari tiga saluran, yaitu Merah (R), Hijau (G), dan Biru (B), masing-masing dengan nilai yang berada dalam interval [255,0,0]. Sementara itu, nilai dalam kubus warna biasanya berkisar pada [1,0,0][3]. Ruang warna RGB merupakan yang paling banyak digunakan karena berdasarkan pencampuran aditif dari tiga warna primer tersebut. Karena ketiga warna primer tersebut bersifat fisik, makna dan aplikasinya pun jelas, sehingga ruang warna RGB sangat sesuai untuk tabung gambar berwarna. Namun demikian, sistem ini tidak sepenuhnya mengakomodasi karakteristik visual manusia. Saat ini, untuk mengukur perbedaan warna dalam ruang RGB umumnya digunakan rumus perbedaan warna berdasarkan jarak dan berdasarkan sudut. Cara kerja algoritma RGB ini melibatkan konversi seluruh gambar ke dalam matriks dua dimensi, dimana ukuran

kolom dan baris ditentukan oleh lebar dan tinggi masing-masing gambar. Setelah gambar dibagi, setiap entri terdiri dari piksel gambar. Nilai RGB yang diambil dari gambar untuk setiap piksel nya adalah 32-bit. Untuk mengekstraksi setiap sub-nilai yaitu merah, hijau, dan biru kita harus menggeser nilai bit yang ada dengan menggeser senilai 24-bit untuk mendapatkan nilai alfa. Identifikasi penyakit berdasarkan ruang warna RGB dapat menghasilkan hasil yang baik karena adanya hubungan yang kuat antar saluran warna. Selain RGB terdapat juga model normalisasi RGB yang mempresentasikan persentase dari sebuah piksel pada citra digital (Waliidaturrahmaniah et al., 2023).

b. *Grayscale Image*

*Grayscale* merujuk pada tingkat keabuan setiap piksel dalam suatu citra, di mana informasi yang disimpan hanya mengenai intensitas hitam dan putih. Citra dikatakan *grayscale* jika tidak mengandung komponen warna merah, hijau, dan biru (RGB); dengan kata lain, citra tersebut hanya memiliki rentang intensitas mulai dari putih, yang merupakan intensitas tertinggi, hingga hitam dengan intensitas terendah, seperti yang ditunjukkan pada Gambar 2.2. Selain itu, citra *grayscale* tersusun atas koordinat spasial x dan y, yang masing-masing memiliki nilai intensitas tertentu. Dalam citra 8-bit, intensitas tiap piksel berkisar antara 0 (hitam) hingga 255 (putih) (Situngkir, 2019).

c. *Threshold image Threshold*

*Threshold* pada citra berfungsi untuk mengubah citra menjadi hanya terdiri dari warna hitam dan putih (2-bit) sehingga tampilannya menjadi lebih jelas. Proses ini melibatkan penentuan nilai ambang batas yang digunakan untuk mengkonversi setiap elemen matriks citra menjadi hitam (0) atau putih (1). Jika nilai piksel berada di bawah ambang batas yang ditetapkan, maka piksel tersebut diubah menjadi hitam (0); sedangkan jika nilainya melebihi ambang batas, maka diubah menjadi putih (1). Untuk citra dengan format RGB, langkah awal yang perlu dilakukan adalah mengonversinya terlebih dahulu menjadi grayscale (citra keabuan) (Situngkir, 2019).



*Gambar 2. 2 Citra Thresholding (Situngkir, 2019)*

#### *d. Augmentasi Citra*

Setelah citra di rubah menjadi citra *grayscale*, kemudiandi lakukan proses data augmentasi. Seperti di ketahui, untuk mendapatkan performa optimal, deep learning membutuhkan data dengan jumlah yang banyak. Karena variasi objek yang ada di lokasi penelitian dan kemampuan pengambilan data cukup terbatas, maka di lakukan data augmentasi untuk memperbanyak variasi data. Data augmentasi adalah sebuah teknik memanipulasi sebuah data tanpa kehilangan inti atau esensi data tersebut.

*Augementasi* citra pada *pyton* bisa di lakukan menggunakan *library* *ImageDataGenerator* yang sudah disediakan oleh keras. Setelah melakukan *import Packege*, fungsi baru bisa *ImageDataGenerator* di panggil,kemudian menentukan augmentasi apa saja yang akan diaplikasikan ke gambar. (Akbar Nugraha et al., 2022)

#### *4. Deep Learning*

*Deep learning*, yang merupakan cabang dari machine learning yang menggunakan jaringan saraf tiruan dengan beberapa lapisan, telah menarik perhatian yang signifikan dalam pengolahan citra digital karena kemampuannya untuk mengekstraksi fitur yang kompleks dan abstrak dari data citra (Akbar Nugraha et al., 2022).

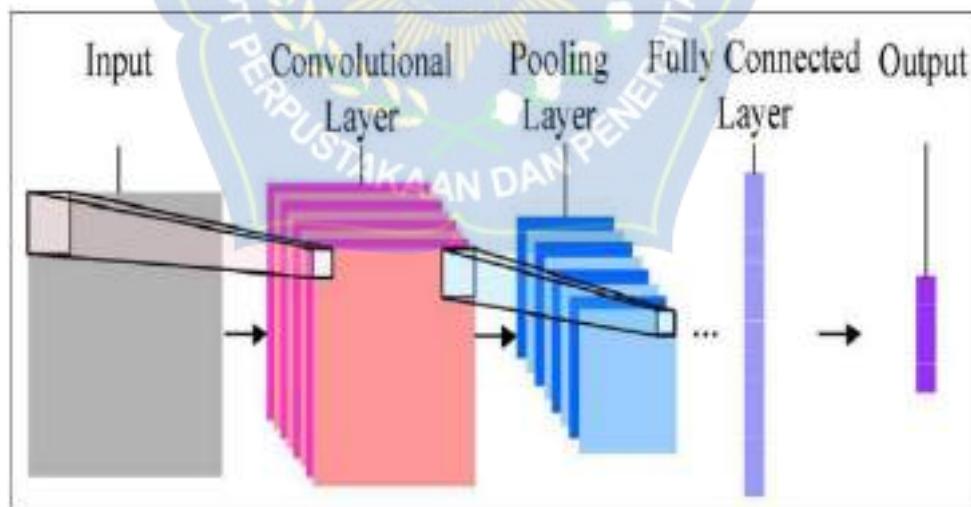
Penelitian ini mengeksplorasi berbagai aspek penggunaan algoritma deep learning dalam konteks pengolahan citra digital, termasuk segmentasi citra, klasifikasi objek, rekonstruksi citra, dan deteksi anomali. Studi ini menguraikan konsep dasar dari algoritma deep learning, dengan penekanan pada arsitektur jaringan saraf konvensional (CNN) yang sering digunakan dalam pengolahan citra digital. Penelitian sebelumnya telah menunjukkan bahwa CNN mampu menghasilkan representasi fitur yang kuat dan efektif dari citra, yang kemudian dapat digunakan untuk berbagai tugas pengolahan citra. Selanjutnya, jurnal ini menyajikan aplikasi praktis dari algoritma deep learning dalam berbagai bidang pengolahan citra digital, termasuk diagnostik medis, pengenalan pola, pemrosesan gambar satelit, dan analisis visual (Kusuma, 2024).

## 5. CNN (*Convolutional Neural Network*)

Menurut penelitian (Situngkir, 2019). CNN salah satu metode yang sering digunakan dalam sistem identifikasi. CNN merupakan pengembangan dari Multi-Layer Perceptron (MLP), yang termasuk dalam jenis *neural network feedforward* (non-rekursif). CNN dirancang khusus untuk memproses data berdimensi dua. Karena kemampuannya dalam menangani jaringan dengan kedalaman tinggi dan penerapannya yang luas dalam pemrosesan citra, CNN disarankan sebagai bagian dari *Deep Neural Network*.

CNN digunakan untuk menganalisis gambar visual serta mendeteksi dan mengenali objek di dalamnya. Algoritma ini termasuk dalam jenis supervised learning, di mana proses kerjanya dimulai dengan menerima input dalam bentuk gambar. CNN sangat cocok untuk pemrosesan dengan input berupa citra. karena pada CNN biasa digunakan untuk mengklasifikasikan citra. CNN adalah suatu metode *machine learning* yang dikembangkan dari *Multi Layer Perceptron* yang didesain untuk mengolah data 2 dimensi (Lesmana et al., 2022).

Secara umum, CNN memiliki komponen seperti bobot (weight), bias, dan fungsi aktivasi (activation function). Tahapan dalam Convolutional Neural Network (CNN) adalah sebagai berikut:



Gambar 2. 3 Arsitektur Convolutional Neural Network (Situngkir, 2019)

Dari gambar 2.4, menerut penelitian (Situngkir, 2019) dapat dilihat ada lima tipe layer dalam *convolutional network*:

*a. ConvolutionLayer*

*Convolution layer* merupakan susunan neuron yang dirancang membentuk filter dengan ukuran tertentu dalam satuan piksel. Biasanya, lapisan pertama dalam tahap ekstraksi fitur adalah lapisan konvolusi dengan ukuran  $5 \times 5 \times 3$ , di mana panjangnya 5 piksel, tingginya 5 piksel, dan ketebalannya 3 sesuai dengan jumlah channel pada gambar..

Pada Gambar 2.4, terdapat tiga filter yang akan digeser ke seluruh bagian gambar. Dalam proses ini, dilakukan operasi dot antara input dengan masing-masing filter, menghasilkan keluaran berupa feature map atau activation map. Dengan menggeser filter ke seluruh area gambar, diperoleh hasil akhir berupa kumpulan feature map yang merepresentasikan fitur-fitur penting dari gambar tersebut. (Situngkir, 2019)

*b. Stride*

*Stride* adalah parameter yang menentukan sejauh mana filter bergeser saat melakukan konvolusi. Jika *stride* ditetapkan sebesar 2, maka filter akan bergeser dua piksel secara horizontal terlebih dahulu, lalu secara vertikal. Besar kecilnya nilai *stride* berpengaruh terhadap waktu komputasi *stride* yang lebih kecil menghasilkan detail yang lebih baik tetapi membutuhkan waktu pemrosesan yang lebih lama. Namun, perlu dicatat bahwa penggunaan *stride* kecil tidak selalu menghasilkan output yang optimal, karena dapat menyebabkan kelebihan informasi yang tidak diperlukan.

### c. Padding

*Padding (zero padding)* adalah parameter yang menentukan jumlah piksel dengan nilai 0 yang ditambahkan ke setiap sisi input. Penambahan zero padding bertujuan untuk mengontrol dimensi output dari feature map, memungkinkan hasil konvolusi memiliki ukuran yang sesuai dengan kebutuhan, serta membantu mempertahankan informasi dari tepi gambar.

Misalnya, jika sebuah layer berukuran  $5 \times 5$  mengalami proses konvolusi menggunakan filter berukuran  $3 \times 3$  dengan stride 2, maka ukuran feature map yang dihasilkan adalah  $2 \times 2$ . Selanjutnya apabila diberikan padding sejumlah 1, maka akan dihasilkan feature map berukuran  $3 \times 3$  dimana telah terjadi manipulasi dimensi dan lebih banyak informasi yang didapat.

### d. Pooling layer

Pergeseran filter dalam proses konvolusi ditentukan oleh dua parameter utama, yaitu *stride* dan *padding*. *Stride* menentukan jumlah piksel yang bergeser baik secara horizontal maupun vertikal. Nilai *stride* yang lebih kecil memungkinkan pengambilan informasi yang lebih detail dibandingkan dengan *stride* yang lebih besar, namun juga meningkatkan waktu komputasi. Meskipun *stride* kecil dapat menangkap lebih banyak detail, hal ini tidak selalu menjamin hasil output yang optimal, karena dapat menyebabkan informasi berlebih atau kurangnya generalisasi dalam model.

*Padding (zero padding)* adalah parameter yang menentukan jumlah piksel bernilai 0 yang akan ditambahkan ke setiap sisi input. Penambahan zero padding bertujuan untuk mengubah dimensi output dari feature map, agar dimensi tersebut tetap sama atau setidaknya hanya mengalami pengurangan yang minimal.

Selanjutnya, dapat digunakan convolutional layer yang lebih dalam untuk menghasilkan feature map dengan ekstraksi fitur yang lebih banyak. Ini juga akan meningkatkan performa model karena informasi dari filter difokuskan pada data yang relevan yang ada pada area zero padding. Untuk menghitung dimensi dari feature map, dapat digunakan persamaan dengan kernel 2x2.

$$\text{output} = \frac{N-F+2P}{S} + 1 \quad (1)$$

Dimana:

$N$  = Panjang/Tinggi Input

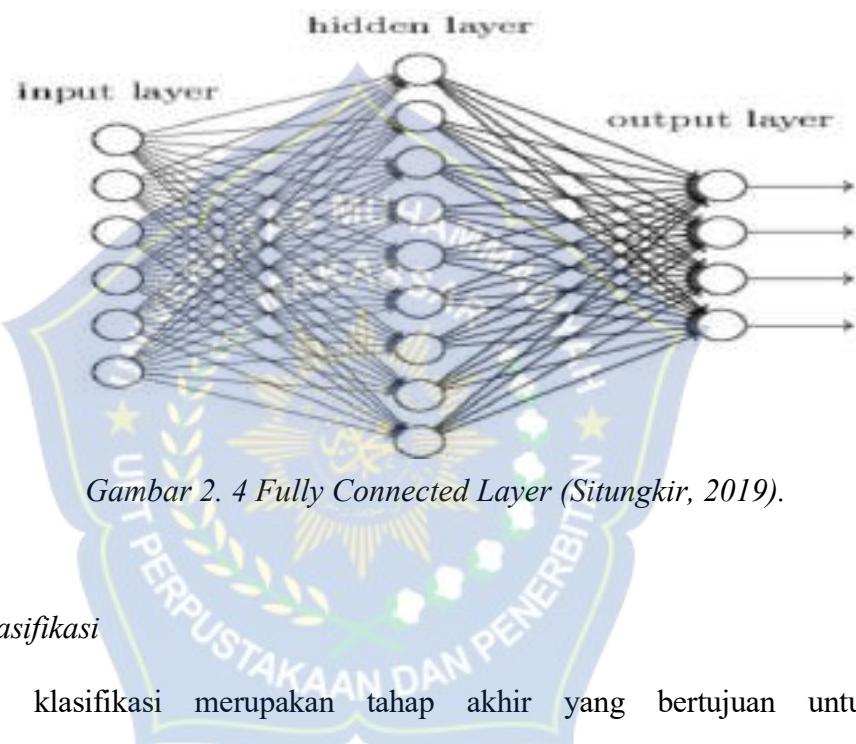
$F$  = Panjang/Tinggi Filter

$P$  = Padding (bernilai 0)

$S$  = Stride

e. *Fully connected layer*

Diperlukan proses pembentukan ulang feature map atau flattening untuk mengubahnya menjadi vektor, karena feature map yang dihasilkan dari ekstraksi fitur masih berupa lapisan multidimensi. Langkah ini dilakukan agar hasil ekstraksi tersebut dapat digunakan sebagai input untuk *Fully Connected Layer*.



Gambar 2. 4 Fully Connected Layer (Situngkir, 2019).

f. *Klasifikasi*

klasifikasi merupakan tahap akhir yang bertujuan untuk menentukan kelas atau label dari data yang telah diproses sebelumnya. Setelah fitur-fitur penting dari input, seperti gambar, diekstraksi melalui lapisan-lapisan konvolusi dan pooling, hasilnya kemudian diratakan melalui proses yang disebut flatten.

Data yang telah diratakan ini kemudian dimasukkan ke dalam *fully connected layer*, yaitu lapisan yang menghubungkan semua neuron dan berfungsi seperti jaringan saraf biasa.

Lapisan ini menganalisis kombinasi dari fitur-fitur yang telah diekstraksi untuk menghasilkan prediksi akhir. Di akhir proses, terdapat output layer yang dilengkapi dengan fungsi aktivasi seperti softmax (untuk klasifikasi multi-kelas) atau sigmoid (untuk klasifikasi biner), yang menghasilkan nilai probabilitas dari masing-masing kelas. Nilai dengan probabilitas tertinggi akan menjadi hasil klasifikasi CNN terhadap input yang diberikan. Tahap klasifikasi ini sangat penting karena merupakan bagian yang menentukan output akhir dari seluruh proses dalam jaringan CNN.

## B. Penelitian Terkait

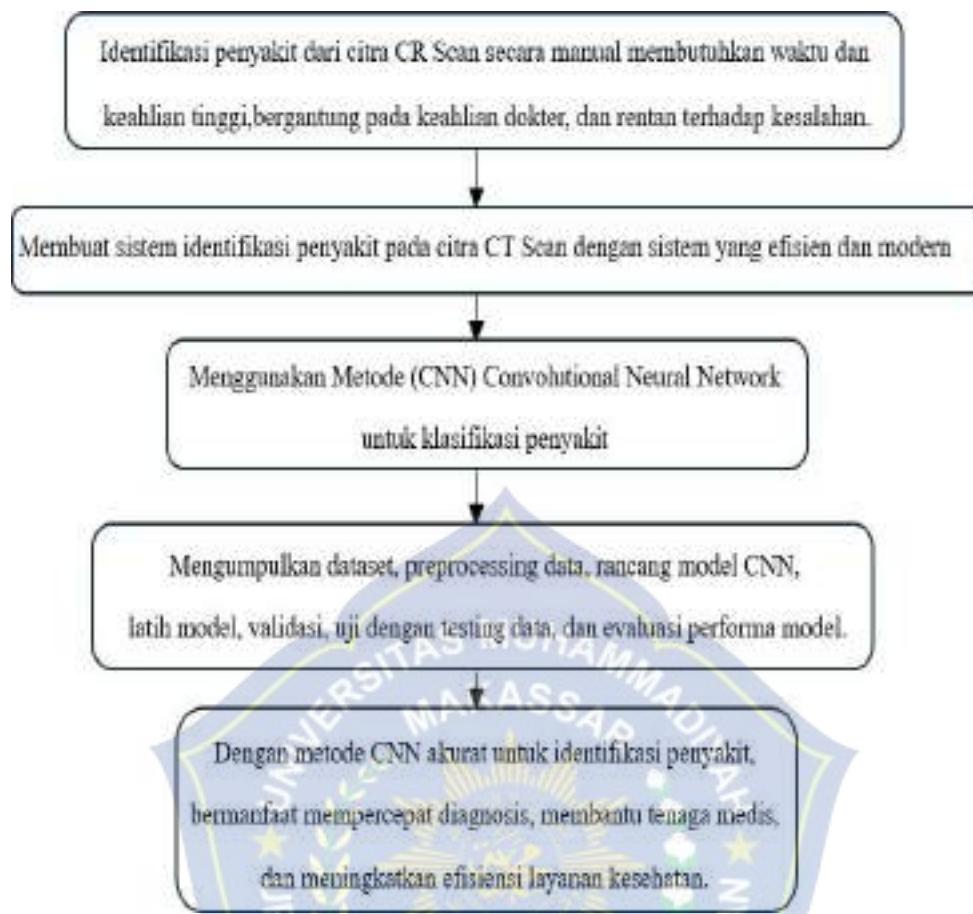
Penelitian	Tempat	Metode	Hasil
Identifikasi penyakit tumor otak pada citra MRI menggunakan metode Convolutional Neural Network (CNN) untuk membantu diagnosis dini.	Dilakukan di Laboratorium Multimediasi dan Pengolahan Citra, Jurusan Informatika, Universitas Islam Negeri Alauddin Makassar	Eksperimen menggunakan dataset citra MRI otak yang diklasifikasikan menjadi dua kategori	Model CNN mampu mengklasifikasikan citra MRI otak dengan akurasi mencapai 94% pada data uji, menunjukkan kinerja yang baik untuk deteksi dini tumor otak.

Mengetahui karakteristik, pola pengobatan, dan efek samping pada lansia dengan stroke iskemik.	RSUP Dr. Hasan Sadikin Bandung (bangsal neurologi).	Deskriptif potong lintang, data rekam medik 2015–2019, <i>simple random sampling</i> , 100 pasien lansia $\geq 60$ tahun.	Mayoritas lansia muda (60–69 tahun), laki-laki, faktor risiko terbanyak hipertensi. Politerapi 79%, polifarmasi 21%. Efek samping 10% pasien, dominan stress ulcer, lebih sering pada polifarmasi.
Identifikasi stroke pada citra CT-scan kepala menggunakan transfer learning CNN (AlexNet, VGG16, GoogLeNet).	FMIPA Universitas Tanjungpura, pemrosesan di Google Colab, data dari Kaggle.	400 citra (200 stroke, 200 normal), dibagi 80% latih dan 20% uji. Tahapan: preprocessing pelatihan model pengujian analisis akurasi.	Akurasi tertinggi 93% pada AlexNet+AdaGrad dan GoogLeNet+RMSProp. Waktu tercepat pelatihan 4 menit 15 detik (AlexNet+AdaGrad).
Pendeteksian penyakit kulit pada manusia menggunakan <i>Image</i>	Jurusan Teknik Elektro, Fakultas Teknik, Universitas Singaperbangsa	Dataset gambar penyakit kulit dikumpulkan dari Google Images, diproses dengan	Model CNN menghasilkan akurasi 99% dengan <i>loss</i> 1,92% dalam mendeteksi penyakit

<i>Recognition</i> berbasis (CNN).	Karawang.	<i>preprocessing,</i> dilatih menggunakan CNN	kulit.
--	-----------	--	--------



### C. Kerangka Pikir



*Gambar 2. 5 Kerangka Pikir*

## **BAB III**

### **METODE NELITIAN**

#### **A. Tempat dan Waktu Penelitian**

##### **1. Tempat Penelitian**

Tempat penelitian adalah tempat di mana penelitian akan dilakukan.

Peneliti melakukan penentuan lokasi karena ini merupakan langkah penting dalam proses penelitian karena memudahkan peneliti untuk melakukan penelitian. Peneliti memilih lokasi penelitian di Rumah Sakit Labuang Baji.

##### **2. Waktu Penelitian**

Adapun pelaksanaan penelitian ini dilakukan jangka waktu kurang lebih 3 bulan, yaitu dimulai pada bulan Januari 2025 sampai dengan Maret 2025.

#### **B. Alat dan Bahan**

Adapun alat dan bahan yang akan digunakan dalam penelitian ini adalah sebagai berikut :

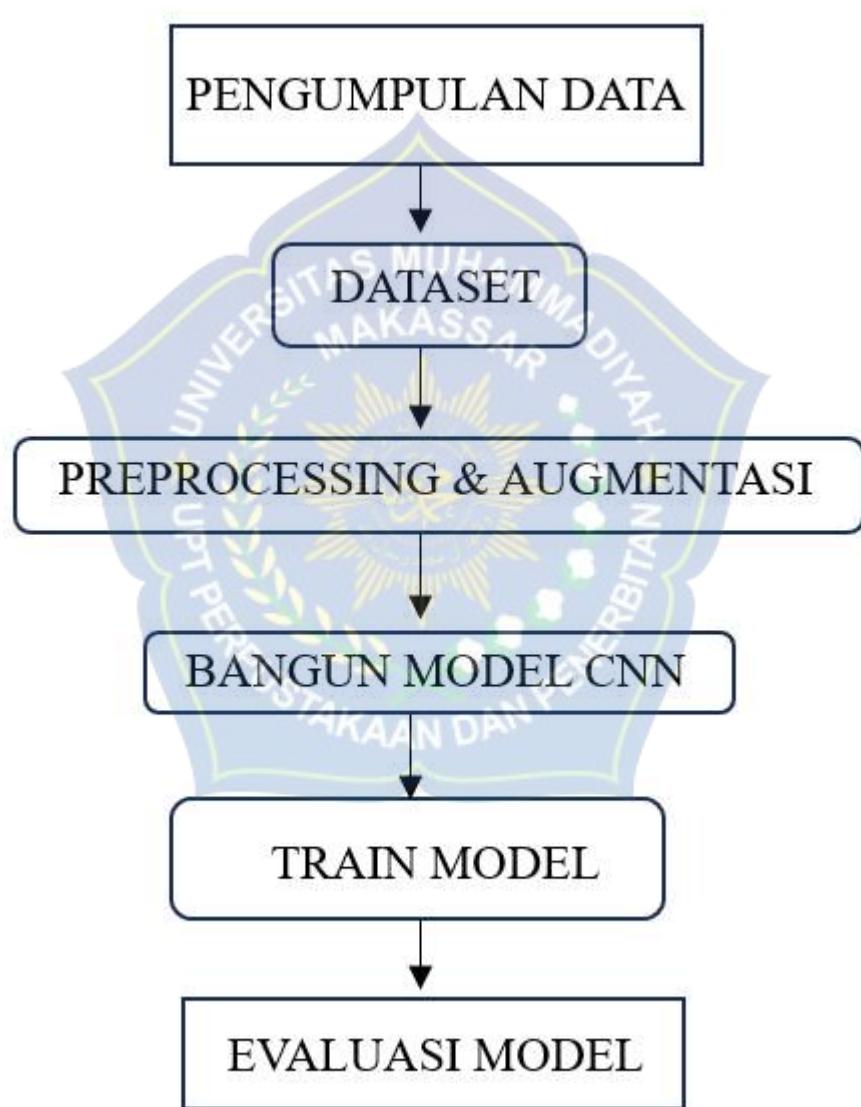
1. Kebutuhan Hardware (perangkat keras)
  - a. Laptop ASUS Vivobook Intel Core i5 - SSD 256 GB
2. Kebutuhan Software (perangkat lunak)
  - a. Windows dan MobileNetV2
  - b. MySQL database
  - c. Python sebagai bahasa programing

#### **C. Perancangan Sistem**

Ketika membangun suatu sistem, perancangan sistem adalah tahap yang sangat penting karena menjelaskan cara membangun fungsi-fungsi yang diperlukan untuk pengoperasian sistem mulai dari tahap perencanaan sistem. Perancangan

sistem juga merupakan langkah dalam pembuatan objek uji, dan tujuan dari perancangan sistem adalah untuk menentukan apakah sistem yang dibuat akan memberikan hasil yang diharapkan.

Berikut flowchart identifikasi penyakit pada CT Scan menggunakan metode Convolutional Neural Network.



*Gambar 3. 1 Flowchart Perancangan Sistem*

## 1. Dataset

Data citra yang digunakan dalam penelitian ini berasal dari citra CT-Scan yang diperoleh dari rumah sakit melalui perangkat pencitraan medis seperti CT scan. Sebanyak 400 Dataset Masing-masing dataset terdiri dari 200 citra normal dan 200 citra yang menunjukkan penyakit. Data tersebut akan digunakan sebagai data pelatihan dan pengujian.

## 2. Desain Sistem

Sebelum sistem pada penelitian ini di bangun, perlu adanya sebuah desain dari sistem yang akan di bangun terlebih dahulu. Desain sistem ini akan menjadi gambaran besar seperti apa jalannya sistem yang akan di bangun nantinya.

Secara sederhana pada penelitian ini, sistem yang akan di bangun nantinya terdiri dari 3 bagian utama, yaitu:

1. *Input* gambar
2. *Proprocesssing Citra*
3. Implementasi CNN
4. *Output*

Seperti yang di jelaskan pada sub bab yasitu data set, data yang sudah dikumpulkan akan menjadi masukan yang akan digunakan. *Dataset* tersebut nantinya akan diproses pada bagian *image processing* algoritma CNN. Baru setelah pemrosesan selesai, akan menghasilkan *output* berupa hasil identifikasi penyakit. Dari bagian utama sistem ini, pada masing masing bagian terdapat pemrosesan sendiri sendiri. Berikut akan dijabarkan pemrosesan yang terjadi pada setiap bagian.

a) Input Gambar

Gambar objek yang sebelumnya sudah dikumpulkan, nantinya akan digunakan sebagai masukan pada sistem yang akan dibangun ini. Masukan gambar pada sistem ini dibagi menjadi 2 masukan, yang pertama masukan gambar sebagai data *training* dan masukan gambar sebagai data *testing*.

b) *Preprocessing Citra*

Sebelum citra dijadikan masukan untuk pelatihan. Citra diolah terlebih dahulu agar lebih memudahkan algoritma CNN untuk melakukan pelatihan dan menentukan ciri dari citra yang dimasukkan.

3. *Augmentasi Citra*

Data augmentasi adalah sebuah teknik memanipulasi sebuah data tanpa kehilangan inti atau esensi data tersebut. Augmentasi yang dilakukan *rescale*, *rotate*, *zoom* dan *flip*.

a. *Rescale Citra* fungsi dari  $rescale=1./255$  adalah untuk menormalisasi citra yaitu dengan membagi piksel kecil dengan piksel terbesar. Fungsi ini akan dikenakan pada semua gambar sebelum melakukan augmentasi dengan fungsi lainnya.

b. *Shear* fungsi dari  $shear\_range=0.2$  adalah untuk menggeser citra

searah jarum jam dengan pergeseran sebanyak 0.2 derajat.

c. *zoom* berfungsi dari  $zoom\_range=0.2$  adalah untuk memperbesar citra dengan perbesaran sebanyak  $1+0.2$  dari luas gambar.

d. *Rotation* berfungsi dari  $rotation = 20$  adalah untuk memutar gambar dengan sudut 30 derajat secara acak.

- e. *Flip* berfungsi dari *horizontal\_flip=True* adalah untuk membalik gambar secara horizontal.

#### 4. Bangun Model *Mobilenetv2*

Model CNN dibangun dengan memanfaatkan arsitektur *MobileNetV2* sebagai inti dari sistem klasifikasi citra. *MobileNetV2* merupakan model deep learning yang ringan namun efisien, yang awalnya dikembangkan untuk klasifikasi gambar pada perangkat dengan keterbatasan sumber daya seperti ponsel atau perangkat edge computing. Meskipun demikian, performanya sangat kompetitif untuk berbagai tugas visi komputer. Dalam konteks ini, *MobileNetV2* digunakan sebagai feature extractor, yaitu untuk mengekstrak fitur penting dari gambar sebelum dilanjutkan ke proses klasifikasi.

#### 5. Klasifikasi.

Dalam penelitian ini, metode yang digunakan untuk proses klasifikasi adalah Convolutional Neural Network (CNN).

Klasifikasi dalam konteks ini merujuk pada proses model CNN dalam mengategorikan citra CT scan ke dalam kelas-kelas tertentu berdasarkan pola yang dikenali. Setiap citra CT scan akan diproses oleh model untuk mengidentifikasi apakah citra tersebut termasuk dalam kategori penyakit Stroke Iskemik.

Model akan memprediksi kelas setiap citra baru berdasarkan fitur yang telah dipelajari, dan hasilnya akan berupa label yang menunjukkan jenis penyakit atau kondisi yang terdeteksi pada citra tersebut.

## 6. *Train Model*

Setelah model CNN berbasis *MobileNetV2* selesai dibangun dan dikompilasi, langkah selanjutnya adalah melatih model (*train model*) menggunakan data pelatihan yang telah diproses sebelumnya. Proses pelatihan dilakukan dengan memanggil fungsi `model.fit()`, yang secara otomatis mengatur alur pelatihan berdasarkan data yang diberikan, jumlah epoch (pengulangan), dan ukuran batch. Selama pelatihan, model akan mempelajari pola-pola dari data pelatihan, yaitu dengan menyesuaikan bobot-bobot internalnya untuk meminimalkan kesalahan prediksi berdasarkan fungsi loss yang telah ditentukan pada tahap kompilasi.

## 7. Evaluasi Model

Setelah model selesai dilatih, langkah selanjutnya adalah evaluasi model, yaitu proses untuk mengukur sejauh mana performa model dalam mengenali data yang belum pernah dilihat sebelumnya. Evaluasi dilakukan dengan menggunakan data validasi atau data uji, yang tidak digunakan selama proses pelatihan, sehingga hasil evaluasi mencerminkan kemampuan model secara objektif dalam melakukan generalisasi terhadap data baru.

Melalui evaluasi ini, kita dapat mengetahui apakah model sudah cukup baik atau masih perlu perbaikan. Misalnya, jika akurasi pada data pelatihan tinggi namun rendah pada data validasi, maka itu menunjukkan bahwa model mengalami *overfitting*, yaitu terlalu menghafal data pelatihan dan gagal mengenali pola baru. Sebaliknya, jika performa pada kedua data hampir sama dan cukup tinggi, itu menandakan bahwa model berhasil belajar secara efektif.

## D. Teknik Pengujian Sistem

Pada proses pengujian sistem ini, metode penelitian ini dilakukan untuk memastikan bahwa model CNN yang dikembangkan mampu mengidentifikasi penyakit pada citra CT Scan secara akurat dan andal dengan menggunakan confusion matrix merupakan alat pengukuran yang dapat digunakan untuk menghitung kinerja atau tingkat kebenaran dari proses klasifikasi. Confusion Matrix dapat menganalisa seberapa baik classifier untuk mengenali dari tiap kelas kelas yang berbeda. (Retno Mutia et al., 2021)

Confusion matrix adalah alat yang digunakan untuk mengevaluasi kinerja model klasifikasi dengan membandingkan prediksi model dengan label yang sebenarnya. Matrix ini memberikan gambaran yang jelas tentang bagaimana model membuat prediksi untuk setiap kelas.

Confusion Matrix adalah tabel yang menggambarkan kinerja model klasifikasi dengan empat komponen utama: True Positive (TP) untuk prediksi positif yang benar, True Negative (TN) untuk prediksi negatif yang benar, False Positive (FP) untuk prediksi positif yang salah, dan False Negative (FN) untuk prediksi negatif yang salah. Matriks ini membantu menghitung metrik seperti akurasi, presisi, recall, dan F1-score untuk mengevaluasi performa model secara menyeluruh.

Metrik evaluasi adalah ukuran yang digunakan untuk menilai kinerja model machine learning. Dalam klasifikasi, metrik utama meliputi akurasi, yang mengukur persentase prediksi benar; presisi, yang menunjukkan ketepatan prediksi positif; recall, yang mengukur kemampuan model mendeteksi semua kasus positif; dan F1-score, yang merupakan rata-rata harmonis antara presisi dan recall.

Dari confusion matrix, berbagai metrik evaluasi dapat dihitung, seperti:

a. *Accuracy*

Accuracy (akurasi) menunjukkan seberapa besar proporsi dari seluruh data yang berhasil diklasifikasikan dengan benar oleh model.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (2)$$

- 1) TP (True Positive): Data sakit yang berhasil diprediksi dengan benar sebagai sakit.
- 2) TN (True Negative): Data normal yang berhasil diprediksi dengan benar sebagai normal.
- 3) FP (False Positive): Data normal yang salah diprediksi sebagai sakit.
- 4) FN (False Negative): Data sakit yang salah diprediksi sebagai normal.

b. *Precision*

$$Precision = \frac{TP}{TP+FP} \quad (3)$$

Precision mengukur seberapa akurat prediksi positif yang dilakukan oleh model, yaitu dari semua prediksi "sakit", berapa banyak yang benar-benar sakit. Precision tinggi berarti model jarang memberikan alarm palsu (false positive), sangat penting untuk menghindari diagnosis salah pada pasien sehat.

c. *Recall*

$$\text{Recall} = \frac{TP}{TP + FN} \quad (4)$$

Recall mengukur kemampuan model dalam mendeteksi semua kasus positif, yaitu dari semua pasien yang benar-benar sakit, berapa yang berhasil dideteksi. Recall tinggi sangat penting dalam bidang medis agar pasien yang sakit tidak sampai terlewatkan (false negative).

d. *F1 – Score*

$$F1 - \text{Score} = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (5)$$

F1-Score merupakan rata-rata harmonis dari precision dan recall. Metrik ini digunakan ketika diperlukan keseimbangan antara kedua metrik tersebut. F1-Score berguna ketika ada ketidakseimbangan antara jumlah data sehat dan sakit, serta saat kesalahan false positive dan false negative sama-sama penting.

Confusion matrix sangat berguna untuk menganalisis jenis kesalahan yang dilakukan oleh model, seperti apakah model lebih sering salah dalam mengklasifikasikan kelas positif atau negatif. Ini membantu dalam memahami area yang perlu diperbaiki dalam model.

Dalam kasus klasifikasi multi-kelas, confusion matrix dapat diperluas untuk mencakup lebih dari dua kelas, dengan masing-masing kelas memiliki baris dan kolom untuk menunjukkan distribusi prediksi yang benar dan salah.

Keuntungan dari confusion matrix adalah kemampuannya untuk memberikan wawasan yang lebih mendalam tentang kinerja model dibandingkan dengan hanya menggunakan akurasi. Namun, confusion matrix juga memiliki keterbatasan dalam menangani dataset yang sangat tidak seimbang, di mana metrik seperti precision dan recall menjadi lebih penting untuk dianalisis.

Confusion matrix adalah alat yang sangat penting dalam evaluasi model klasifikasi karena memberikan informasi yang lebih rinci tentang kinerja model, yang dapat membantu dalam memperbaiki dan mengoptimalkan algoritma.

## E. Teknik Analisis Data

1. Pengambilan Data
  - a. Sumber data diambil dari rumah sakit melalui alat pencitraan medis seperti CT scan.
  - b. Format data biasanya berupa gambar dalam format JPEG/PNG, disertai metadata riwayat penyakit pasien.
2. Preprocessing Data
  - a. Normalisasi skala nilai paksel citra ke rentang tertentu, misalnya [0,1], untuk memastikan konsistensi data.
  - b. Segmentasi di pisdahkan area penting, seperti organ atau jaringan target, menggunakan algoritma seperti *U-Net* atau metode *thresholding*.
  - c. Augmentasi data menambahkan variasi seperti *rotasi*, *flipping*, atau *zoom* untuk meningkatkan keragaman data mencegah overfitting.

### 3. Feature Extraction

- a. Manual ekstraksi fitur seperti tekstur, bentuk, atau intensitas menggunakan algoritma seperti *GLCM (Gray Level Co-occurrence Matrix)*.
- b. Otomatis menggunakan metode *deep learning* seperti CNN untuk secara otomatis mengekstrak fitur penting dari citra.

### 4. Modeling

- a. Pelatihan model menggunakan dataset terlabel untuk melatih model klasifikasi penyakit *Stroke Iskemik*.
- b. Validasi menggunakan data validasi untuk memastikan model tidak overfitting.
- c. *Parameter optimazation* di lakukan *tuning hyperparameter* untuk meningkatkan performa model.

### 5. Evaluasi Model

Hitung metrix evaluasi seperti akurasi, presisi, recal, F1-score untuk menilai performa model.

### 6. Identifikasi Penyakit

- a. Inferensi menerapkan model pada citra baru untuk memprediksi kemungkinan penyakit.
- b. Visualisasi Hasil menggunakan teknik *Grad-CAM* untuk menyoroti area penting pada citra yang digunakan model untuk membuat prediksi.
- c. Klasifikasi akhir dari hasil prediksi dikategorikan ke dalam kelas penyakit tententu atau kondisi sehat.

7. Interpretasi dan Validasi Klinis

- a. Hasil prediksi dikaji ulang oleh tenaga medis untuk validasi.
- b. Model diintegrasikan ke dalam sistem klinis sebagai alat bantu diagnosis,bukan pengganti tenaga medis.



## BAB IV

### HASIL DAN PEMBAHASAN

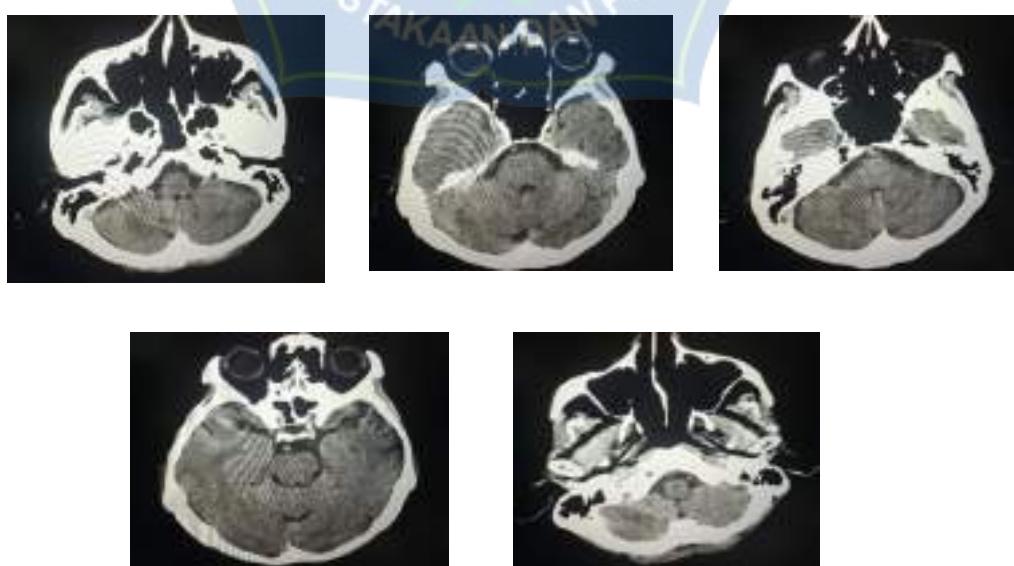
#### A. Hasil dan Gambaran Umum Data

Dataset yang digunakan dalam penelitian ini terdiri dari 400 gambar CT scan, yang terbagi menjadi dua kategori, yaitu:

1. Citra CT scan dari pasien dengan kondisi normal

Dataset yang digunakan dalam penelitian ini terdiri dari 200 gambar CT scan kepala pasien dengan kondisi normal yang diperoleh dari instalasi radiologi Rumah Sakit Labuang Baji.

Gambar-gambar tersebut diambil menggunakan perangkat CT scan medis standar rumah sakit dan telah melalui proses seleksi untuk memastikan bahwa citra yang digunakan tidak menunjukkan indikasi adanya kelainan atau gangguan pada jaringan otak. Dataset ini digunakan sebagai representasi dari kondisi otak sehat untuk membandingkan dan mengidentifikasi perbedaan fitur visual dengan citra CT scan yang menunjukkan gejala penyakit *stroke Iskemik*.

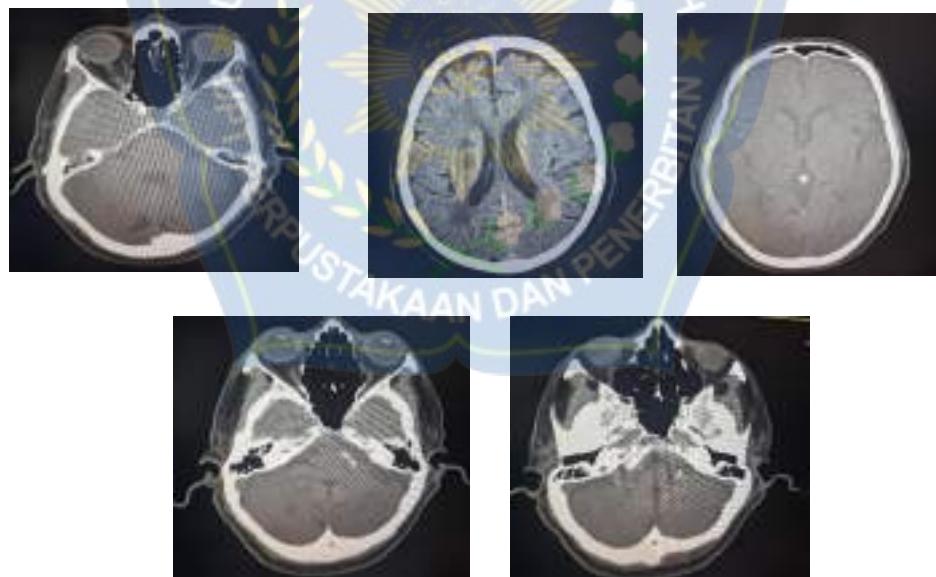


*Gambar 4. 1Dataset CT Scan Normal*

## 2. Citra CT scan dari pasien dengan kondisi sakit Storke Iskemik

Dataset citra CT scan dari pasien dengan kondisi sakit berjumlah 200 gambar dan diperoleh dari instalasi radiologi Rumah Sakit Labuang Baji. Gambar ini mencakup kasus seperti stroke iskemik yang telah terverifikasi oleh tenaga medis.

Setiap gambar diklasifikasikan dalam kategori “sakit” dan digunakan untuk melatih serta menguji model dalam mengenali pola citra otak yang abnormal. Data ini membantu membedakan citra otak sehat dan bermasalah dalam proses klasifikasi menggunakan metode CNN. Setiap gambar memiliki resolusi yang disesuaikan dengan input model. Data ini digunakan sebagai dasar dalam pelatihan dan pengujian model klasifikasi.



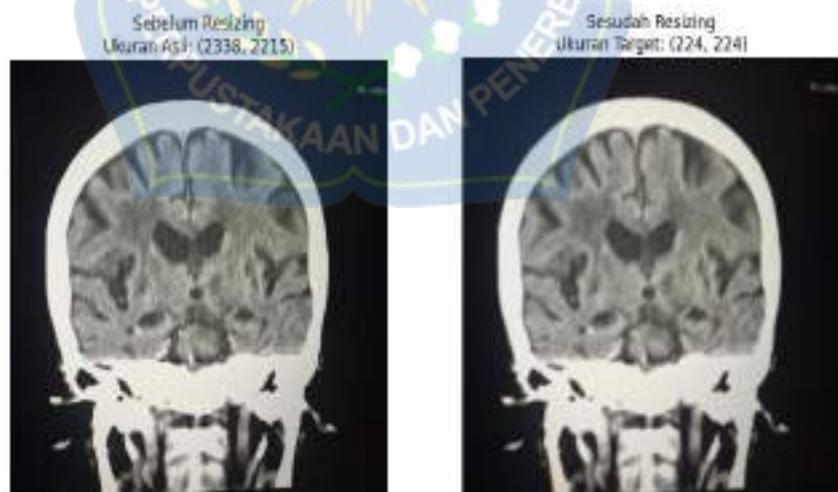
*Gambar 4. 2 Dataset CT Scan Sakit Stroke Iskemik*

## B. Tahap Proprecessing

Sebelum digunakan dalam proses pelatihan, data gambar telah melalui tahap preprocessing dengan langkah-langkah sebagai berikut:

### 1. *Resizing*

Resizing adalah proses mengubah ukuran citra menjadi dimensi yang sesuai untuk diproses oleh model deep learning. Dalam proyek ini, citra otak awalnya berukuran  $2338 \times 2215$  piksel, kemudian diubah menjadi  $224 \times 224$  piksel. Ukuran ini dipilih karena merupakan ukuran standar input untuk model *MobileNetV2*, sebuah arsitektur *pretrained CNN* yang dirancang khusus untuk efisiensi di perangkat dengan sumber daya terbatas. *MobileNetV2* hanya dapat menerima input citra berukuran tetap, yaitu  $224 \times 224$  piksel, agar fitur-fitur penting dapat dikenali dengan optimal oleh layer-layer konvolusinya.



Gambar 4. 3 Tahap Proprecessing Resizing

## 2. Normalisasi

Proses normalisasi dilakukan menggunakan fungsi `preprocess_input` dari modul `tensorflow.keras.applications.mobilenet_v2`. Fungsi ini secara otomatis mengubah skala nilai piksel citra dari rentang 0–255 menjadi -1 hingga 1, sesuai dengan standar input yang digunakan saat model *MobileNetV2* dilatih. Normalisasi ini bertujuan untuk menyesuaikan distribusi data input agar sejalan dengan distribusi data pelatihan awal model, sehingga performa prediksi menjadi lebih akurat dan stabil.

## 3. Label Encoding

Label encoding adalah proses untuk mengubah label kategori dalam bentuk teks (misalnya: *Normal*, *Sakit*) menjadi bentuk angka (numerik). Model pembelajaran mesin seperti CNN tidak dapat memproses data berupa teks secara langsung, sehingga diperlukan transformasi label ke dalam bentuk angka.

Tabel 4.1 Label Encoding

LABEL TEKS	LABEL ANGKA
Normal	0
Sakit Stroke Iskemik	1

Dengan menggunakan metode ini, model dapat membedakan antara kelas-kelas target dalam bentuk nilai numerik, yang kemudian digunakan dalam proses pelatihan dan evaluasi.

## C. Implementasi Model CNN

Model CNN di rancang untuk mengklasifikasikan citra CT Scan menjadi dua kelas, yaitu Normal dan Sakit. Citra yang di gunakan di proses dengan teknik augmentasi seperti *rotasi*, *flipping*, dan *zoom* menggunakan *Image Data Generator*, serta di normalisasi agar nilai piksel berada dalam rentang 0 hingga 1.

Arsitektir CNN terdiri dari tiga lapisan konvolusi (*Conv2D*) dan *poling*, diikuti oleh *Global Average Poling 2D*, satu layer dense dengan aktivasi ReLU, serta *Dropout* untuk mencegah overfitting, Output layer menggunakan satu neuron dengan aktivasi sigmoid untuk menghasilkan prediksi biner. Model dikompilasi dengan *binary\_crossentropy* dan *Adam optimizer*, lalu dilatih selama beberapa epoch menggunakan data pelatihan dan validasi dengan data uji.

### 1. Arsitektur Model CNN

Model CNN yang digunakan merupakan hasil fine-tuning dari arsitektur MobileNetV2, yang kemudian ditambahkan beberapa layer tambahan seperti:

#### a. *GlobalAveragePooling2D*

GlobalAveragePooling2D merangkum fitur dari tiap channel menjadi satu nilai rata-rata. Ini membuat model lebih ringan dan efisien. Dibandingkan dengan Flatten, metode ini menghasilkan parameter lebih sedikit dan meminimalkan risiko overfitting.

b. *Dense Layer* dengan *ReLU*

Layer Dense digunakan setelah pooling agar model dapat belajar dari fitur yang telah diekstrak. Fungsi aktivasi *ReLU* (*Rectified Linear Unit*) membantu mempercepat proses pelatihan dan mengatasi masalah vanishing gradient.

c. Output Layer dengan satu neuron dan fungsi aktivasi sigmoid

Karena klasifikasi yang dilakukan bersifat biner (Normal atau Sakit), digunakan satu neuron output dengan fungsi aktivasi sigmoid. Fungsi sigmoid menghasilkan nilai probabilitas antara 0 dan 1, sehingga cocok untuk dua kelas.

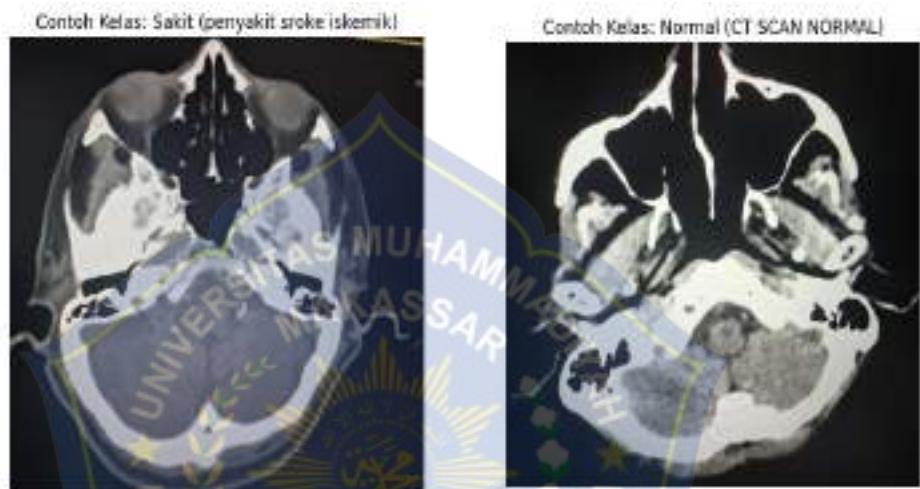
d. *Hyperparameter* Model MobileNetV2

Arsitektur MobileNetV2 dengan bobot pralatin dari ImageNet dan ditambahkan lapisan klasifikasi untuk dua kelas. Pelatihan dilakukan selama 10 *epoch* dengan batch size 32, menggunakan *optimizer Adam* (learning rate 0.0001) dan *loss function binary crossentropy*. Augmentasi gambar seperti rotasi, zoom, dan flip horizontal digunakan untuk menambah variasi data dan mencegah overfitting.

## 2. Analisis Implementasi Model CNN

Implementasi model CNN dalam program ini menunjukkan bahwa jaringan berhasil dibangun secara bertahap dengan struktur lapisan yang umum digunakan, mulai dari *Conv2D*, *MaxPooling2D*, hingga *GlobalAveragePooling2D* dan *Dense*. Arsitektur ini dirancang untuk

mengenali fitur-fitur penting dari citra CT-Scan otak dan mengklasifikasikannya ke dalam dua kategori, yaitu Normal dan Sakit. Penggunaan *ImageDataGenerator* untuk augmentasi data membantu memperkaya variasi input yang diberikan ke model, sehingga model memiliki ketahanan terhadap perubahan kecil pada citra.



Gambar 4. 4 Identifikasi Sakit Stroke Iskemik Dan Normal

Pada gambar diatas menampilkan dua hasil identifikasi dari citra CT scan otak yang diklasifikasikan ke dalam dua kelas berbeda, yaitu kelas sakit (*stroke iskemik*) dan kelas normal.

Pada gambar sebelah kiri, ditunjukkan contoh citra dari pasien dengan kondisi stroke iskemik, yaitu kondisi ketika aliran darah ke otak terganggu akibat penyumbatan pembuluh darah. Citra ini menunjukkan adanya area dengan kerapatan berbeda (hipodens) yang menjadi indikasi adanya

kerusakan jaringan otak akibat kekurangan oksigen. Karakteristik ini bisa dikenali oleh model deteksi berbasis CNN sebagai tanda dari kondisi "sakit".

Sementara itu, gambar di sebelah kanan merupakan contoh CT scan normal, di mana tidak tampak kelainan pada jaringan otak. Struktur otak terlihat simetris dan tidak terdapat area abnormal, sehingga model klasifikasi mengenali gambar ini sebagai "normal".

Perbandingan kedua gambar ini menjadi dasar pelatihan model klasifikasi medis untuk membedakan kondisi otak yang sehat dan yang mengalami gangguan, seperti stroke iskemik.

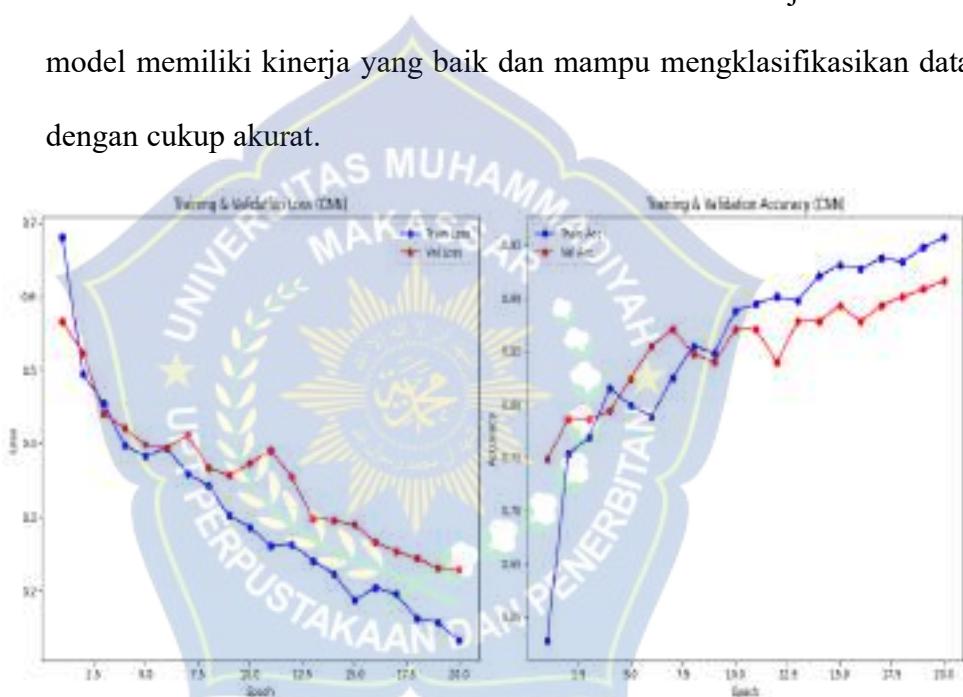
### 3. Visualisasi Loss dan Akurasi Model CNN

Visualisasi loss dan akurasi pada model CNN yang digunakan dalam kode berfungsi untuk memantau kinerja model selama proses pelatihan. Grafik akurasi menunjukkan seberapa baik model mengklasifikasikan data dengan benar dari waktu ke waktu (per *epoch*), sementara grafik loss menggambarkan seberapa besar kesalahan prediksi yang dilakukan model. Pada kode tersebut, visualisasi dibuat menggunakan library Matplotlib, dengan menampilkan grafik *accuracy* dan *val\_accuracy* untuk membandingkan akurasi pada data latih dan validasi, serta *loss* dan *val\_loss* untuk melihat penurunan kesalahan selama pelatihan.

Dengan grafik ini, pengguna dapat mengevaluasi apakah model mengalami overfitting, underfitting, atau berjalan dengan baik, sehingga membantu dalam pengambilan keputusan untuk penyempurnaan model.

### a) Grafik Training dan Loss

Gambar dibawah menunjukkan grafik loss dan akurasi model CNN selama 20 epoch. Grafik loss (kiri) menunjukkan penurunan nilai kesalahan pada data latih dan validasi, menandakan model belajar dengan baik tanpa *overfitting* signifikan. Grafik akurasi (kanan) memperlihatkan peningkatan akurasi secara konsisten, dengan akurasi pelatihan mencapai lebih dari 95% dan validasi sekitar 91%. Hal ini menunjukkan bahwa model memiliki kinerja yang baik dan mampu mengklasifikasikan data dengan cukup akurat.



Gambar 4. 5 Grafik Training dan loss

#### 1. Tujuan Grafik Training dan loss

Grafik akurasi dan loss digunakan untuk melihat kinerja model. Akurasi menunjukkan ketepatan prediksi, sedangkan loss menunjukkan tingkat kesalahan. Akurasi naik dan loss turun menandakan model belajar dengan baik.

## 2. Grafik Akurasi Model

Grafik menunjukkan dua garis:

1. Biru (*Train Accuracy*): Akurasi terhadap data pelatihan
2. Merah (*Val Accuracy*): Akurasi terhadap data validasi

Pada gambar 4.5 grafik di atas menunjukkan perkembangan akurasi dan loss selama proses pelatihan model CNN. Garis biru menunjukkan hasil pada data pelatihan, sementara garis merah menunjukkan hasil pada data validasi. Terlihat bahwa akurasi model meningkat dan loss menurun seiring bertambahnya *epoch*, baik pada data pelatihan maupun validasi. Ini menandakan bahwa model belajar dengan baik dan tidak mengalami *overfitting*.

## D. Hasil Pengujian Model

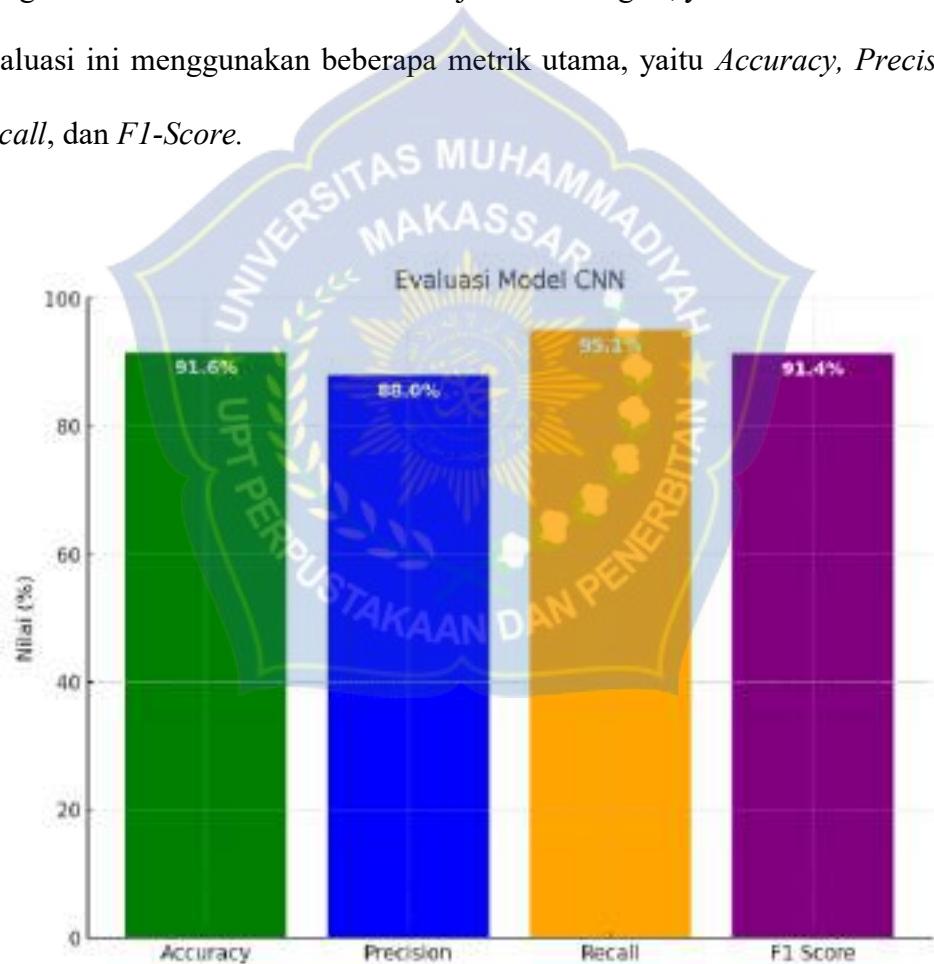
Hasil pengujian model pada kode CNN dengan arsitektur MobileNetV2 menunjukkan bahwa model mampu melakukan klasifikasi citra CT scan otak secara efektif. Setelah proses pelatihan selesai, model diuji menggunakan data uji (*testing*) yang belum pernah dilihat sebelumnya. Evaluasi dilakukan dengan menggunakan metrik akurasi, *precision*, *recall*, dan *f1-score* yang dihasilkan melalui *classification\_report*.

Hasilnya menunjukkan nilai akurasi dan *f1-score* yang cukup tinggi, yang menandakan bahwa model mampu membedakan antara citra otak normal dan yang mengalami *stroke iskemik* dengan baik. Selain itu, penggunaan confusion

matrix memperjelas jumlah prediksi benar dan salah untuk masing-masing kelas, sehingga membantu dalam menilai kemampuan model secara keseluruhan. Secara umum, hasil pengujian ini membuktikan bahwa model cukup handal untuk digunakan dalam tugas klasifikasi medis berbasis citra.

#### a. Evaluasi Model Confusion Matrix

Evaluasi model dilakukan untuk mengukur kinerja model CNN dalam mengklasifikasikan citra CT-Scan menjadi dua kategori, yaitu Normal dan Sakit. Evaluasi ini menggunakan beberapa metrik utama, yaitu *Accuracy*, *Precision*, *Recall*, dan *F1-Score*.



Gambar 4. 6 Performa Hasil Evaluasi Model

### 1. *Accuracy* (91.60%)

Mengukur persentase prediksi yang benar dari seluruh data. Digunakan untuk melihat kinerja model secara keseluruhan. Misalnya, jika dari 100 gambar, 90 diklasifikasikan dengan benar, maka akurasi adalah 90%.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} = \frac{59+61}{59+61+8+3} = \frac{120}{131} = 0.916$$

### 2. *Precision* (88.0%)

Mengukur ketepatan model dalam memprediksi kelas positif. Artinya, dari semua data yang diprediksi "sakit", berapa banyak yang benar-benar sakit. Presisi penting untuk menghindari kesalahan prediksi positif palsu.

$$Precision = \frac{TP}{TP+FP} = \frac{59}{59+8} = \frac{59}{67} = 0.880$$

### 3. *Recall* (95.1%)

Mengukur kemampuan model dalam menemukan semua data yang benar-benar positif. Misalnya, dari semua pasien sakit, berapa banyak yang berhasil dikenali model. Recall penting dalam bidang medis agar tidak ada kasus sakit yang terlewat.

$$Recall = \frac{TP}{TP+FN} = \frac{59}{59+3} = \frac{59}{62} = 0.951$$

#### 4. *F1 Score* (91.4%)

Menggabungkan precision dan recall dalam satu nilai rata-rata harmonis. F1-score berguna saat ingin menyeimbangkan antara ketepatan dan kelengkapan, terutama jika data tidak seimbang antara kelas normal dan sakit.

$$F1\ Score = 2 \times \frac{Precision \times Recall}{precision+recall} = 2 \times \frac{0.880 \times 0.951}{0.880 + 0.951} = 0.914$$

Model yang digunakan memiliki hasil evaluasi yang cukup baik dengan nilai akurasi sebesar 92,5%, yang menunjukkan bahwa sebagian besar prediksi model sudah tepat. *Precision* sebesar 94,7% menandakan bahwa dari seluruh data yang diprediksi sebagai sakit, hampir semuanya benar-benar sakit. *Recall* sebesar 90% menunjukkan kemampuan model dalam mengenali sebagian besar kasus sakit secara benar, sehingga minim risiko terlewatnya pasien yang benar-benar sakit. Sementara itu, *F1 Score* sebesar 92,2% mencerminkan keseimbangan antara *precision* dan *recall*, yang penting terutama jika jumlah data antara kelas normal dan sakit tidak seimbang.

Selain itu, model juga diuji menggunakan *confusion matrix* yang menunjukkan jumlah prediksi benar dan salah:

1. *TP (True Positive)* : 59
2. *TN (True Negative)* : 61
3. *FP (False Positive)* : 8
4. *FN (False Negative)* : 3

Tabel 4.2 *Confusion Matrix*

	Prediksi: Positif (Sakit)	Prediksi: Negatif (Normal)
Aktual: Positif (Sakit)	TP = 59	FN = 3
Aktual: Negatif (Normal)	FP = 8	TN = 61

Pada tabel *Confusion Matrix* diatas menggambarkan performa model klasifikasi dalam membedakan antara kondisi sakit (*stroke iskemik*) dan normal berdasarkan hasil prediksi dan kondisi aktual. Dari total data uji, model berhasil mengklasifikasikan 59 data sakit dengan benar sebagai sakit (*True Positive*), dan 61 data normal dengan benar sebagai normal (*True Negative*). Namun, terdapat 3 data sakit yang salah diklasifikasikan sebagai normal (*False Negative*), serta 8 data normal yang salah diklasifikasikan sebagai sakit (*False Positive*). Hasil ini



Gambar 4. 7 *Confusion Matrix*

menunjukkan bahwa model memiliki tingkat ketepatan yang tinggi dalam mengidentifikasi kedua kelas, dengan kesalahan klasifikasi yang tergolong rendah, sehingga layak digunakan untuk membantu proses diagnosis awal.

Model ini sudah mampu membedakan gambar normal dan sakit dengan cukup baik. Presisi menunjukkan bahwa sebagian besar prediksi positif memang benar, sedangkan recall menunjukkan model berhasil mengenali banyak data positif. Nilai F1-score menunjukkan keseimbangan antara presisi dan recall, sehingga model cukup seimbang dalam membuat prediksi. Namun, kinerjanya masih bisa ditingkatkan agar hasilnya lebih optimal.

### b. Analisi Kesalahan

Model CNN menunjukkan performa baik, namun masih terdapat kesalahan klasifikasi, yaitu 8 data normal diprediksi sebagai sakit (False Positive) dan 3 data sakit diprediksi sebagai normal (False Negative). Kesalahan ini berdampak pada nilai precision dan recall. False negative lebih berisiko karena dapat menyebabkan keterlambatan penanganan pada pasien yang benar-benar sakit. Kesalahan ini kemungkinan disebabkan oleh kemiripan fitur pada gambar CT, kualitas data, atau jumlah data yang tidak seimbang. Untuk memperbaiki, dapat dilakukan peningkatan kualitas dataset, augmentasi data, dan penggunaan teknik interpretasi model seperti *Grad-CAM*.

Analisis Kesalahan: Sistem Memprediksi "penyakit stroke iskemik"  
(Kenyataannya: "CT SCAN NORMAL") - False Positive



Gambar 4. 8 Analisis Kesalahan False Positive

Analisis Kesalahan: Sistem Memprediksi "CT SCAN NORMAL"  
(Kenyataannya: "penyakit stroke iskemik") - False Negative



Gambar 4. 9 Analisis Kesalahan False Negatif

Pada gambar diatas menunjukkan bahwa sistem salah memprediksi gambar CT scan otak. Sistem mengira pasien terkena stroke iskemik, padahal sebenarnya hasil CT scan normal. Kesalahan ini disebut *False Positive*, yaitu ketika sistem menyatakan seseorang sakit padahal tidak. Hal ini bisa terjadi karena citra terlihat mirip dengan kasus stroke atau model belum cukup akurat mengenali perbedaannya.

Pada gambar diatas menunjukkan sebuah kasus analisis kesalahan dalam sistem prediksi medis yang menggunakan CT *scan* otak. Sistem tersebut keliru memprediksi bahwa CT *scan* yang ditampilkan adalah "NORMAL", padahal pada kenyataannya pasien tersebut didiagnosis menderita "penyakit stroke iskemik". Kesalahan ini dikategorikan sebagai "*False Negative*", yang berarti sistem gagal mendeteksi adanya kondisi medis yang sebenarnya ada. Dalam konteks diagnosis medis, *false negative* merupakan jenis kesalahan yang sangat krusial karena dapat menyebabkan penundaan atau ketiadaan penanganan untuk kondisi serius.

## BAB V

### PENUTUP

#### A. Kesimpulan

Berdasarkan hasil pengujian dan analisis yang ada di atas, dapat di simpulkan sebagai berikut:

1. Arsitektur CNN MobileNetV2 digunakan untuk mengidentifikasi stroke iskemik pada citra CT-scan. Model dilengkapi layer tambahan (*GlobalAveragePooling2D*, *Dense ReLU*, dan *sigmoid output*). Data CT-scan normal dan stroke diproses dengan resize  $224 \times 224$  piksel, normalisasi -1 hingga 1, serta augmentasi (rotasi, zoom, flipping) sebelum pelatihan.
2. Model menunjukkan performa yang sangat baik dalam klasifikasi, dengan hasil evaluasi yang tinggi dan kesalahan klasifikasi yang minimal. Setelah proses pelatihan dan pengujian, model CNN menghasilkan akurasi sebesar 91,6%, *precision* sebesar 88%, dan *recall* mencapai 95,1%. Nilai-nilai tersebut menunjukkan bahwa model sangat andal dalam mengenali citra stroke iskemik secara benar (*recall* tinggi) dan cukup tepat dalam prediksi positif (*precision*).

#### B. Saran

Model MobileNetV2 dapat ditingkatkan dengan menambah variasi dan jumlah data untuk mencegah *overfitting*. *Data augmentation* dapat diperluas dengan penyesuaian kecerahan dan penambahan noise. Evaluasi model sebaiknya menggunakan *k-fold cross validation* agar hasil lebih stabil, serta teknik interpretabilitas seperti Grad-CAM untuk menampilkan area citra yang memengaruhi prediksi. Untuk penggunaan klinis, model perlu diuji pada data lebih besar dan divalidasi oleh tenaga medis.

## DAFTAR PUSTAKA

- Akbar Nugraha, F., Ayu Fazira, H., Iqbal Nugraha, M., Febriansyah, A., Setiya Putra, Y. W., Adhim, M. F., Mubarok, H., Andri Nugraha Ramdhon, Fadly Febriya, ADITIYA, D. N., NUGRAHA, C., PRASSETIYO, H., Saputra, I. G., Ibrahim, I., Tanuwijaya, E., & Roseanne, A. (2022). Identifikasi Ekspresi Wajah Berbasis Citra Menggunakan Algoritma Convolutional Neural Network (CNN). *Jurnal Inovasi Teknologi Terapan*, 16(1), 191198. <https://eproceeding.itenas.ac.id/index.php/fti/article/view/1718%0Ahttps://eproceeding.itenas.ac.id/index.php/fti/article/download/1718/1463>
- Dastur, D. P. (2023). Identifikasi Penyakit Tuberculosis Melalui Hasil Citra X-RAY Menggunakan Deep Learning Convolutional Neural Network [Universitas Sriwijaya]. *InRepository.Unsri.Ac.Id*. [https://repository.unsri.ac.id/129112/3/RAMA\\_20201\\_03041281924041\\_0030078404\\_01\\_front\\_ref.pdf](https://repository.unsri.ac.id/129112/3/RAMA_20201_03041281924041_0030078404_01_front_ref.pdf)
- Febriyanti, F. A. (2024). Image Processing Dengan Metode Convolutional Neural Network (Cnn) Untuk Deteksi Penyakit Kulit Pada Manusia. *Kohesi: Jurnal Sains Dan Teknologi*, 3(10), 21–30. <https://ejournal.warunayama.org/kohesi>
- Hisni, D., Saputri, M. E., & Sujarni, S. (2022). Faktor - Faktor Yang Berhubungan Dengan Kejadian Stroke Iskemik Di Instalasi Fisioterapi Rumah Sakit Pluit Jakarta Utara Periode Tahun 2021. *Jurnal Penelitian Keperawatan Kontemporer*, 2(1), 140–149. <https://doi.org/10.59894/jpkk.v2i1.333>
- Kusuma, T. (2024). Studi Algoritma Deep Learning Dalam Pengolahan Citra Digital. *Jurnal Dunia Data*, 1(5), 1–15. <http://www.pusdansi.org/index.php/duniadata/article/view/108>

Lesmana, A. M., Fadhillah, R. P., & Rozikin, C. (2022). Identifikasi Penyakit pada Citra Daun Kentang Menggunakan Convolutional Neural Network (CNN). *Jurnal Sains Dan Informatika*, 8(1), 21–30. <https://doi.org/10.34128/jsi.v8i1.377>

Retno Mutia, D., Kumalasari, N., Pratiwi, C., & Fu'adah, Y. N. (2021). Perancangan Identifikasi Penyakit Pada Retina Berbasis Pengolahan Citra Optical Coherence Tomography (Oct) Menggunakan Convolutional Neural Network (Cnn) Identification of Diseases in Retina Based on Optical Coherence Tomography (Oct) Image Using Convolut. *ISSN: 2355-9365 e-Proceeding of Engineering*, 8(6), 11551–11559. [www.kaggle.com](http://www.kaggle.com),

Sholihany, R. F., Waluyo, A., & Irawati, D. (2021). Latihan Rom Pasif Unilateral Dan Bilateral Terhadap Peningkatan Kekuatan Otot Akibat Stroke Iskemik. *Keperawatan Silampari*, 4(17), 399–405.

Situngkir, G. R. (2019). *Identifikasi Penyakit Tumor Otak Pada Citra Ct-Scan*. Universitas Sumatera Utara Medan.

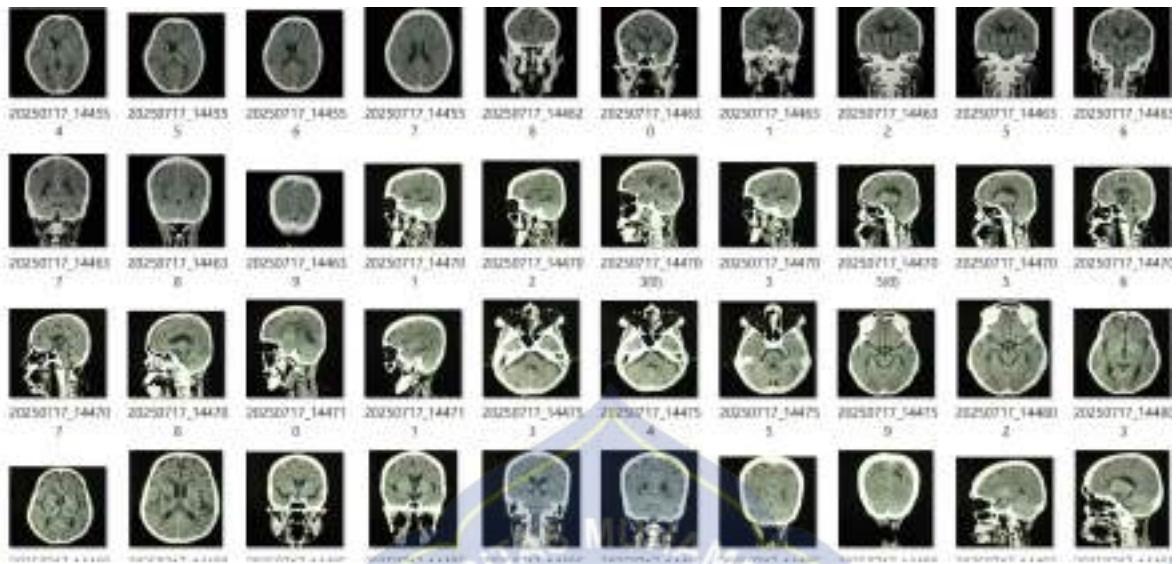
Tamburian, A. G., Rataq, B. T., & Nelwan, J. E. (2020). Hubungan antara Hipertensi, Diabetes Melitus, dan Hipercolesterolemia dengan Kejadian Stroke Iskemik. *Journal of Public Health and Community Medicine*, 1(1), 27–33.

Togu, G. M., Lisda Amalia, & Trully Deti Rose Sitorus. (2021). Pola Pengobatan Stroke Iskemik Pada Pasien Lansia di Rumah Sakit Hasan Sadikin Bandung. *Journal Of The Indonesian Medical Association*, 71(2), 65–70. <https://doi.org/10.47830/jimma-vol.71.2-2021-387>

Waliidaturrahmaniah, W., Hasanuddin, H., & Wahyuni, D. (2023). Identifikasi Stroke Menggunakan Metode Transfer learning Arsitektur Convolutional Neural Network Pada Citra CT-scan Kepala. *Prisma Fisika*, 11(3), 78. <https://doi.org/10.26418/pf.v11i3.65242>

## LAMPIRAN

Lampiran 1. Dataset Mentah Penyakit Stroke Iskemik dan Normal



Gambar 1. Penyakit Stroke Iskemik



Gambar 2. Normal

Lampiran 2. *Source Code*

```
import torch
import torchvision.transforms as transforms
from torchvision.datasets import ImageFolder
from torch.utils.data import DataLoader, random_split
from torch import nn, optim
import os
import matplotlib.pyplot as plt
from sklearn.metrics import classification_report, confusion_matrix,
ConfusionMatrixDisplay
from sklearn.model_selection import StratifiedShuffleSplit
import cv2
from PIL import Image
import numpy as np
import pandas as pd
from datetime import datetime
import ipywidgets as widgets
from IPython.display import display, clear_output
import io
import seaborn as sns

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

drive_path='/content/drive/My Drive/'

DATASET_PATH = "/content/drive/MyDrive/Data penyakit (skripsi)"
BATCH_SIZE = 16
EPOCHS = 20
LR = 1e-4
IMG_SIZE = 224
VAL_SPLIT = 0.3
CONFIDENCE_THRESHOLD = 0.70

device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
print(f'Menggunakan perangkat: {device}')

# Path untuk menyimpan model CNN terbaik
BEST_CNN_MODEL_PATH = "best_cnn_ct_model.pth"
```

```

from sklearn.model_selection import StratifiedShuffleSplit
from torch.utils.data import Subset

# 1. Transformasi Data
transform_train = transforms.Compose([
    transforms.Resize((IMG_SIZE, IMG_SIZE)),
    transforms.RandomHorizontalFlip(p=0.5),
    transforms.RandomRotation(10),
    transforms.ToTensor(),
    transforms.Normalize([0.5, 0.5, 0.5], [0.5, 0.5, 0.5])
])

# Transformasi untuk validasi (tanpa augmentasi)
transform_val = transforms.Compose([
    transforms.Resize((IMG_SIZE, IMG_SIZE)),
    transforms.ToTensor(),
    transforms.Normalize([0.5, 0.5, 0.5], [0.5, 0.5, 0.5])
])

# 2. Memuat Dataset dari Folder
train_ds_source = ImageFolder(root=DATASET_PATH, transform=transform_train)
val_ds_source = ImageFolder(root=DATASET_PATH, transform=transform_val)

# Secara dinamis mendapatkan nama dan jumlah kelas dari folder.
class_names = train_ds_source.classes
num_classes = len(class_names)
print(f"Kelas yang ditemukan: {class_names}")
print(f"Jumlah kelas: {num_classes}")

# 3. Membagi Dataset dengan Stratifikasi (Menjaga proporsi kelas)
targets = train_ds_source.targets
sss = StratifiedShuffleSplit(n_splits=1, test_size=VAL_SPLIT, random_state=42)
train_indices, val_indices = next(sss.split(np.zeros(len(targets)), targets))

# Buat subset dari sumber data yang SESUAI
train_dataset = Subset(train_ds_source, train_indices)
val_dataset = Subset(val_ds_source, val_indices)

print(f"Jumlah data training: {len(train_dataset)}, Jumlah data validasi: {len(val_dataset)}")

# 4. Membuat DataLoader
train_loader = DataLoader(train_dataset, batch_size=BATCH_SIZE, shuffle=True,
num_workers=2)
val_loader = DataLoader(val_dataset, batch_size=BATCH_SIZE, shuffle=False,
num_workers=2)

```

```

class TraditionalCNN(nn.Module):
    def __init__(self, num_classes):
        super(TraditionalCNN, self).__init__()
        self.conv1 = nn.Conv2d(3, 32, kernel_size=3, padding=1)
        self.relu1 = nn.ReLU()
        self.pool1 = nn.MaxPool2d(kernel_size=2, stride=2)

        self.conv2 = nn.Conv2d(32, 64, kernel_size=3, padding=1)
        self.relu2 = nn.ReLU()
        self.pool2 = nn.MaxPool2d(kernel_size=2, stride=2)

        self.conv3 = nn.Conv2d(64, 128, kernel_size=3, padding=1)
        self.relu3 = nn.ReLU()
        self.pool3 = nn.MaxPool2d(kernel_size=2, stride=2)

        self.flatten = nn.Flatten()
        # Perhitungan ukuran input untuk fc1: 128 channels * (IMG_SIZE/8) *
        (IMG_SIZE/8)
        # IMG_SIZE = 224 -> 224/2=112 -> 112/2=56 -> 56/2=28. Jadi 128 * 28 * 28
        self.fc1 = nn.Linear(128 * 28 * 28, 512)
        self.relu4 = nn.ReLU()
        self.dropout = nn.Dropout(0.5)
        self.fc2 = nn.Linear(512, num_classes)

    def forward(self, x):
        x = self.pool1(self.relu1(self.conv1(x)))
        x = self.pool2(self.relu2(self.conv2(x)))
        x = self.pool3(self.relu3(self.conv3(x)))
        x = self.flatten(x)
        x = self.dropout(self.relu4(self.fc1(x)))
        x = self.fc2(x)
        return x

# Inisialisasi model CNN, criterion, dan optimizer
cnn_model = TraditionalCNN(num_classes=num_classes).to(device)
cnn_criterion = nn.CrossEntropyLoss()
cnn_optimizer = optim.Adam(cnn_model.parameters(), lr=LR)
cnn_history = {"train_loss": [], "train_acc": [], "val_loss": [], "val_acc": []}
best_cnn_val_accuracy = 0.0

@torch.no_grad()
def evaluate_cnn(epoch):
    cnn_model.eval()
    y_true, y_pred = [], []
    total_val_loss = 0
    for inputs, labels in val_loader:

```

```

inputs, labels = inputs.to(device), labels.to(device)
outputs = cnn_model(inputs)
loss = cnn_criterion(outputs, labels)
total_val_loss += loss.item()

_, preds = torch.max(outputs, 1)
y_true.extend(labels.cpu().numpy())
y_pred.extend(preds.cpu().numpy())

acc = sum(1 for yt, yp in zip(y_true, y_pred) if yt == yp) / len(y_true)
val_loss = total_val_loss / len(val_loader)

print("\n--- Laporan Validasi CNN ---")
print(f"Validation Loss: {val_loss:.4f}")
print(f"Validation Accuracy: {acc:.2%}")

# Classification Report
print("\n--- Classification Report ---")
print(classification_report(y_true, y_pred, target_names=class_names, zero_division=0))

# Confusion Matrix
print("\n--- Confusion Matrix ---")
cm = confusion_matrix(y_true, y_pred)

# Asumsi kelas 0: Geger Otak (concussion), kelas 1: Normal
# Anda bisa menyesuaikan ini jika urutan kelasnya berbeda
tn, fp, fn, tp = cm.ravel()

print(f"True Negatives (TN) [Prediksi Normal, Aktual Normal]: {tn}")
print(f"False Positives (FP) [Prediksi Geger Otak, Aktual Normal]: {fp}")
print(f"False Negatives (FN) [Prediksi Normal, Aktual Geger Otak]: {fn}")
print(f"True Positives (TP) [Prediksi Geger Otak, Aktual Geger Otak]: {tp}")

# Visualisasi Confusion Matrix
plt.figure(figsize=(8, 6))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=class_names,
            yticklabels=class_names)
plt.xlabel('Predicted Label')
plt.ylabel('True Label')
plt.title(f'Confusion Matrix - Epoch {epoch+1}')
plt.savefig(f'confusion_matrix_epoch_{epoch+1}.png')
plt.show()

return acc, val_loss

```

```

def plot_cnn_training_history():
    epochs_range = range(1, len(cnn_history['train_loss']) + 1)
    plt.figure(figsize=(15, 6))

    plt.subplot(1, 2, 1)
    plt.plot(epochs_range, cnn_history["train_loss"], 'bo-', label="Train Loss")
    plt.plot(epochs_range, cnn_history["val_loss"], 'ro-', label="Val Loss")
    plt.xlabel("Epoch")
    plt.ylabel("Loss")
    plt.title("Training & Validation Loss (CNN)")
    plt.legend()

    plt.subplot(1, 2, 2)
    plt.plot(epochs_range, cnn_history["train_acc"], 'bo-', label="Train Acc")
    plt.plot(epochs_range, cnn_history["val_acc"], 'ro-', label="Val Acc")
    plt.xlabel("Epoch")
    plt.ylabel("Accuracy")
    plt.title("Training & Validation Accuracy (CNN)")
    plt.legend()

    plt.tight_layout()
    plt.savefig("training_history_cnn.png")
    plt.show()

def train_cnn():
    global best_cnn_val_accuracy

    print("\n\n--- MEMULAI TRAINING MODEL CNN TRADISIONAL ---")
    for epoch in range(EPOCHS):
        cnn_model.train()
        total_loss, correct, total = 0, 0, 0

        print(f"\n--- Epoch {epoch+1}/{EPOCHS} (CNN) ---")
        for i, (inputs, labels) in enumerate(train_loader):
            inputs, labels = inputs.to(device), labels.to(device)
            cnn_optimizer.zero_grad()
            outputs = cnn_model(inputs)
            loss = cnn_criterion(outputs, labels)
            loss.backward()
            cnn_optimizer.step()

            total_loss += loss.item()
            _, preds = torch.max(outputs, 1)
            correct += (preds == labels).sum().item()
            total += labels.size(0)
    
```

```

print(f"\rEpoch {epoch+1}/{EPOCHS}, Batch {i+1}/{len(train_loader)}", end="")

train_loss = total_loss / len(train_loader)
train_acc = correct / total

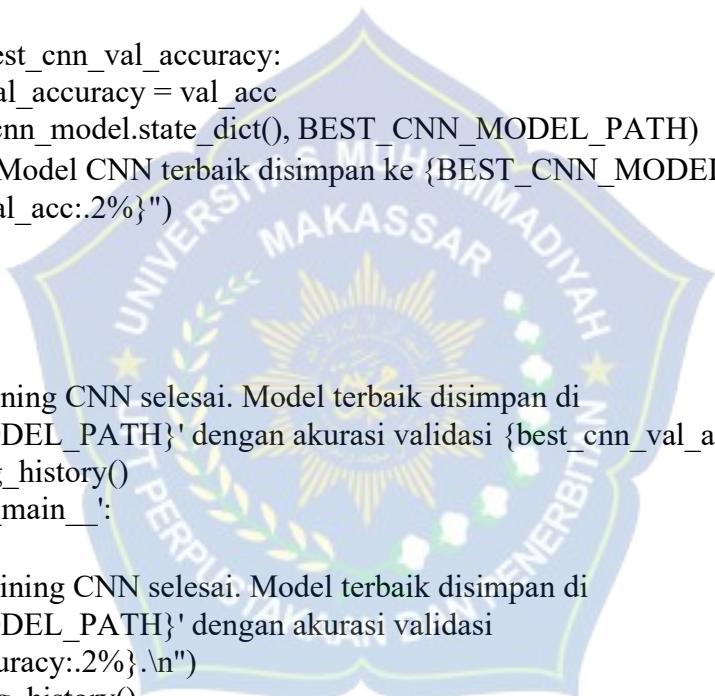
val_acc, val_loss = evaluate_cnn(epoch)

cnn_history["train_loss"].append(train_loss)
cnn_history["train_acc"].append(train_acc)
cnn_history["val_loss"].append(val_loss)
cnn_history["val_acc"].append(val_acc)

print(f"\nEpoch {epoch+1} Summary (CNN) - Train Loss: {train_loss:.4f} - Train
Acc: {train_acc:.4f} - Val Loss: {val_loss:.4f} - Val Acc: {val_acc:.4f}")

if val_acc > best_cnn_val_accuracy:
    best_cnn_val_accuracy = val_acc
    torch.save(cnn_model.state_dict(), BEST_CNN_MODEL_PATH)
    print(f" ✅ Model CNN terbaik disimpan ke {BEST_CNN_MODEL_PATH}
dengan akurasi: {val_acc:.2%}")

```



```

# train_cnn()
# print(f"\n ✅ Training CNN selesai. Model terbaik disimpan di
'{BEST_CNN_MODEL_PATH}' dengan akurasi validasi {best_cnn_val_accuracy:.2%}.")
# plot_cnn_training_history()
if __name__ == '__main__':
    train_cnn()
    print(f"\n ✅ Training CNN selesai. Model terbaik disimpan di
'{BEST_CNN_MODEL_PATH}' dengan akurasi validasi
{best_cnn_val_accuracy:.2%}.")
    plot_cnn_training_history()

```

```

# --- 4. Definisi & Training Model MobileNetV2 ---
from torchvision.models import mobilenet_v2, MobileNet_V2_Weights

mobilenet_model = mobilenet_v2(weights=MobileNet_V2_Weights.DEFAULT)
num_ftrs = mobilenet_model.classifier[1].in_features
mobilenet_model.classifier[1] = nn.Linear(num_ftrs, num_classes)
mobilenet_model = mobilenet_model.to(device)

mobilenet_criterion = nn.CrossEntropyLoss()
mobilenet_optimizer = optim.Adam(mobilenet_model.parameters(), lr=LR)
mobilenet_history = {"train_loss": [], "train_acc": [], "val_loss": [], "val_acc": []}
best_mobilenet_val_accuracy = 0.0

```

```

# Path untuk menyimpan model MobileNetV2 terbaik (menambahkan ini juga)
BEST_MOBILENET_MODEL_PATH = "best_mobilenet_ct_model.pth"

@torch.no_grad()
def evaluate_mobilenet(epoch):
    mobilenet_model.eval()
    y_true, y_pred = [], []
    total_val_loss = 0
    for inputs, labels in val_loader:
        inputs, labels = inputs.to(device), labels.to(device)
        outputs = mobilenet_model(inputs)
        loss = mobilenet_criterion(outputs, labels)
        total_val_loss += loss.item()
        _, preds = torch.max(outputs, 1)
        y_true.extend(labels.cpu().numpy())
        y_pred.extend(preds.cpu().numpy())

    acc = sum(1 for yt, yp in zip(y_true, y_pred) if yt == yp) / len(y_true)
    val_loss = total_val_loss / len(val_loader)

    print("\n--- Laporan Validasi MobileNetV2 ---")
    print(f"Validation Loss: {val_loss:.4f}, Validation Accuracy: {acc:.2%}")

    print("\n--- Classification Report (MobileNetV2) ---")
    print(classification_report(y_true, y_pred, target_names=class_names, zero_division=0))

    cm = confusion_matrix(y_true, y_pred)
    tn, fp, fn, tp = cm.ravel()
    print("\n--- Confusion Matrix (MobileNetV2) ---")
    print(f"True Negatives (TN): {tn}, False Positives (FP): {fp}")
    print(f"False Negatives (FN): {fn}, True Positives (TP): {tp}")

    plt.figure(figsize=(8, 6))
    sns.heatmap(cm, annot=True, fmt='d', cmap='Greens', xticklabels=class_names,
    yticklabels=class_names)
    plt.xlabel('Predicted Label')
    plt.ylabel('True Label')
    plt.title(f"Confusion Matrix MobileNetV2 - Epoch {epoch+1}")
    plt.savefig(f"confusion_matrix_mobilenet_epoch_{epoch+1}.png")
    plt.show()

    return acc, val_loss

```

```

def train_mobilenet():
    global best_mobilenet_val_accuracy
    print("\n\n--- MEMULAI TRAINING MODEL MOBILENETV2 ---")
    for epoch in range(EPOCHS):
        mobilenet_model.train()
        total_loss, correct, total = 0, 0, 0
        print(f"\n--- Epoch {epoch+1}/{EPOCHS} (MobileNetV2) ---")
        for i, (inputs, labels) in enumerate(train_loader):
            inputs, labels = inputs.to(device), labels.to(device)
            mobilenet_optimizer.zero_grad()
            outputs = mobilenet_model(inputs)
            loss = mobilenet_criterion(outputs, labels)
            loss.backward()
            mobilenet_optimizer.step()
            total_loss += loss.item()
            _, preds = torch.max(outputs, 1)
            correct += (preds == labels).sum().item()
            total += labels.size(0)
            print(f"\rEpoch {epoch+1}/{EPOCHS}, Batch {i+1}/{len(train_loader)}", end="")

        train_loss = total_loss / len(train_loader)
        train_acc = correct / total
        val_acc, val_loss = evaluate_mobilenet(epoch)

        mobilenet_history["train_loss"].append(train_loss)
        mobilenet_history["train_acc"].append(train_acc)
        mobilenet_history["val_loss"].append(val_loss)
        mobilenet_history["val_acc"].append(val_acc)

        print(f"\nEpoch {epoch+1} Summary (MobileNetV2) - Train Loss: {train_loss:.4f} - Train Acc: {train_acc:.4f} - Val Loss: {val_loss:.4f} - Val Acc: {val_acc:.4f}")

        if val_acc > best_mobilenet_val_accuracy:
            best_mobilenet_val_accuracy = val_acc
            torch.save(mobilenet_model.state_dict(), BEST_MOBILENET_MODEL_PATH)
            print(f" ✅ Model MobileNetV2 terbaik disimpan ke {BEST_MOBILENET_MODEL_PATH} dengan akurasi: {val_acc:.2%}")

# --- 5. Fungsi Plot Perbandingan ---
def plot_comparison_error_rate(cnn_hist, mobilenet_hist):
    epochs_range = range(1, EPOCHS + 1)

    cnn_val_error = [1 - acc for acc in cnn_hist['val_acc']]
    mobilenet_val_error = [1 - acc for acc in mobilenet_hist['val_acc']]

```

```

plt.figure(figsize=(12, 8))
plt.plot(epochs_range, cnn_val_error, 'ro-', label='CNN Baseline Error Rate')
plt.plot(epochs_range, mobilenet_val_error, 'go-', label='MobileNetV2 Error Rate')

plt.title('Perbandingan Tingkat Kesalahan Validasi: CNN vs MobileNetV2', fontsize=16)
plt.xlabel('Epoch', fontsize=12)
plt.ylabel('Tingkat Kesalahan (1 - Akurasi)', fontsize=12)
plt.xticks(epochs_range)
plt.legend(fontsize=12)
plt.grid(True)
plt.savefig("comparison_error_rate.png")
plt.show()

```

```

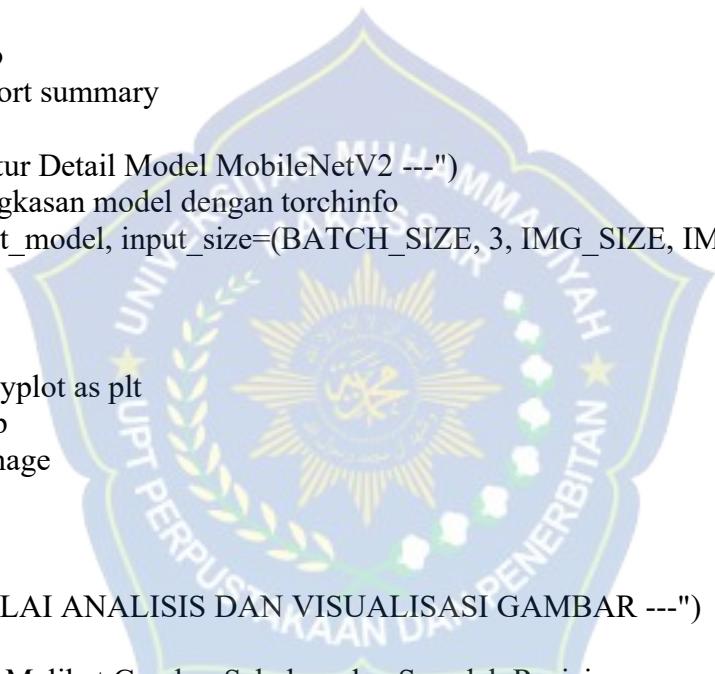
pip install torchinfo
from torchinfo import summary

```

```

print("\n--- Arsitektur Detail Model MobileNetV2 ---")
# Menampilkan ringkasan model dengan torchinfo
summary(mobilenet_model, input_size=(BATCH_SIZE, 3, IMG_SIZE, IMG_SIZE))

```



```

import torch
import matplotlib.pyplot as plt
import numpy as np
from PIL import Image
import random

print("\n--- MEMULAI ANALISIS DAN VISUALISASI GAMBAR ---")

# --- 1. Kode untuk Melihat Gambar Sebelum dan Sesudah Resizing ---
print("\n1. Menampilkan contoh gambar sebelum dan sesudah resizing...")

# Ambil satu gambar acak dari dataset validasi
original_img_path, _ = random.choice(val_ds_source.samples)
original_img = Image.open(original_img_path)

# Definisikan transform untuk resizing
resize_transform = transforms.Resize((IMG_SIZE, IMG_SIZE))
resized_img = resize_transform(original_img)

# Tampilkan kedua gambar
fig, axs = plt.subplots(1, 2, figsize=(12, 6))
axs[0].imshow(original_img)
axs[0].set_title(f'Sebelum Resizing\nUkuran Asli: {original_img.size}')
axs[0].axis('off')

```

```

    axs[1].imshow(resized_img)
    axs[1].set_title(f'Sesudah Resizing\nUkuran Target: ({IMG_SIZE}, {IMG_SIZE})')
    axs[1].axis('off')
    plt.savefig("before_after_resize.png")
    plt.show()

# --- 2. Kode untuk Menampilkan Gambar Identifikasi Normal dan Sakit ---
print("\n2. Menampilkan contoh gambar untuk setiap kelas...")

# Cari path gambar untuk masing-masing kelas menggunakan class_names
path_sakit = ""
path_normal = ""

# Assuming class_names[0] is 'CT SCAN NORMAL' and class_names[1] is 'penyakit
stroke iskemik'
# based on the output of cell a2Hh4mheu6Cz
normal_class_name = class_names[0]
sakit_class_name = class_names[1]

for img_path, label_idx in train_ds_source.samples:
    if class_names[label_idx] == sakit_class_name and not path_sakit:
        path_sakit = img_path
    elif class_names[label_idx] == normal_class_name and not path_normal:
        path_normal = img_path
    if path_sakit and path_normal:
        break

# Buka dan tampilkan gambar
img_sakit = Image.open(path_sakit)
img_normal = Image.open(path_normal)

fig, axs = plt.subplots(1, 2, figsize=(12, 6))
axs[0].imshow(img_sakit)
axs[0].set_title(f'Contoh Kelas: Sakit ({sakit_class_name})')
axs[0].axis('off')

axs[1].imshow(img_normal)
axs[1].set_title(f'Contoh Kelas: Normal ({normal_class_name})')
axs[1].axis('off')
plt.savefig("sample_per_class.png")
plt.show()

# --- 3. Kode untuk Analisis Kesalahan (False Positive & False Negative) ---
# Menggunakan model MobileNetV2 yang memiliki akurasi lebih baik
print("\n3. Menganalisis kesalahan prediksi dari model MobileNetV2...")

```

```

# Pastikan model dalam mode evaluasi
mobilenet_model.load_state_dict(torch.load(BEST_MOBILENET_MODEL_PATH))
mobilenet_model.to(device)
mobilenet_model.eval()

# Kumpulkan semua prediksi dan label asli dari data validasi
y_true_all, y_pred_all, all_indices = [], [], []
with torch.no_grad():
    for inputs, labels in val_loader:
        inputs_gpu = inputs.to(device)
        outputs = mobilenet_model(inputs_gpu)
        _, preds = torch.max(outputs, 1)

        y_true_all.extend(labels.cpu().numpy())
        y_pred_all.extend(preds.cpu().numpy())
    # Kita perlu menyimpan index asli dari val_dataset
    # DataLoader tidak mengembalikan index, jadi kita buat manual
    # Ini akan bekerja jika shuffle=False di val_loader
    start_index = len(all_indices)
    end_index = start_index + len(labels)
    # Need to map the batch indices back to the original dataset indices
    batch_original_indices = [val_dataset.indices[i] for i in range(start_index, end_index)]
    all_indices.extend(batch_original_indices)

# Konversi ke array numpy untuk kemudahan
y_true_all = np.array(y_true_all)
y_pred_all = np.array(y_pred_all)
all_indices = np.array(all_indices)

# Cari index kelas
label_sakit = class_names.index(sakit_class_name)
label_normal = class_names.index(normal_class_name)

# 3a. Analisis Kesalahan Sistem: Prediksi Sakit, Nyata Normal (False Positive)
fp_indices = np.where((y_pred_all == label_sakit) & (y_true_all == label_normal))[0]

if len(fp_indices) > 0:
    # Ambil satu contoh acak
    fp_index_in_batch = random.choice(fp_indices)
    original_dataset_index = all_indices[fp_index_in_batch]
    img_path, _ = val_ds_source.samples[original_dataset_index]
    fp_img = Image.open(img_path)

    plt.figure(figsize=(6, 6))
    plt.imshow(fp_img)
    plt.title(f'Analisis Kesalahan: Sistem Memprediksi')

```

```

" {sakit_class_name}\n(Kenyataannya: "{normal_class_name}") - False Positive')
plt.axis('off')
plt.savefig("false_positive_example.png")
plt.show()
else:
    print(f"Tidak ditemukan kesalahan False Positive ({sakit_class_name} diprediksi,
{normal_class_name} nyata) pada data validasi.")

# 3b. Analisis Kesalahan Sistem: Prediksi Normal, Nyata Sakit (False Negative)
fn_indices = np.where((y_pred_all == label_normal) & (y_true_all == label_sakit))[0]

if len(fn_indices) > 0:
    # Ambil satu contoh acak
    fn_index_in_batch = random.choice(fn_indices)
    original_dataset_index = all_indices[fn_index_in_batch]
    img_path, _ = val_ds_source.samples[original_dataset_index]
    fn_img = Image.open(img_path)

    plt.figure(figsize=(6, 6))
    plt.imshow(fn_img)
    plt.title(f'Analisis Kesalahan: Sistem Memprediksi
{normal_class_name}\n(Kenyataannya: "{sakit_class_name}") - False Negative')
    plt.axis('off')
    plt.savefig("false_negative_example.png")
    plt.show()
else:
    print(f"Tidak ditemukan kesalahan False Negative ({normal_class_name} diprediksi,
{sakit_class_name} nyata) pada data validasi.")

print("\n--- Analisis dan Visualisasi Gambar Selesai ---")

```

### Lampiran 3. Permohonan Penelitian Kepada Ketua Program Studi Informatika



Lampiran 4. Permohonan Penelitian Kepada LP3M Universitas Muhammadiyah Makassar

**MAJELIS PENDIDIKAN TINGGI PIMPINAN PUSAT MUHAMMADIYAH  
UNIVERSITAS MUHAMMADIYAH MAKASSAR  
FAKULTAS TEKNIK  
PROGRAM STUDI INFORMATIKA**

Nomor : 017/05/IF/C 4-VI/III/46/2025  
Lamp. : -  
Hai : Permohonan Data Penelitian

Kepada yang Terhormat,  
**Ketua LP3M Unismuh Makassar**

Di -  
Tempat

*Assalamu 'Alaikum Warahmatullahi Wabarakatuh*

Dengan Rahmat Allah SWT, Semoga aktivitas kita bermilai ibadah di Sisi-Nya. Dalam rangka penyelesaian Tugas Akhir pada Program Studi Informatika dengan judul "Identifikasi Penyakit Pada Citra CT Scan Menggunakan Metode Convolutional Neural Network". Bersama ini kami sampaikan mahasiswa:

Stambuk	Nama
105 84 11113 20	Agung Faturrahman

Seshubungan dengan hal tersebut, maka kami mohon dibantuan surat pengantar pada instansi di bawah ini:

Nama Instansi : RSUD Labueng Baji Prov. Sulsel  
Alamat : Jln. DR. Ratulenga No.81, Labueng Baji, Kec. Mamajang, Kota Makassar, Sulawesi Selatan.

Demikian surat kami atas perhatian dan kerja samanya kami haturkan baik-baik terima kasih.

*Jazakumullah Khaeran Katsiran*  
*Wassalamu 'Alaikum warahmatullahi Wabarakatuh*

*Ketua Program Studi  
Informatika*  
*Muhammad A.M Havat, S.Kom., M.T.*  
*NIP. 641087901*

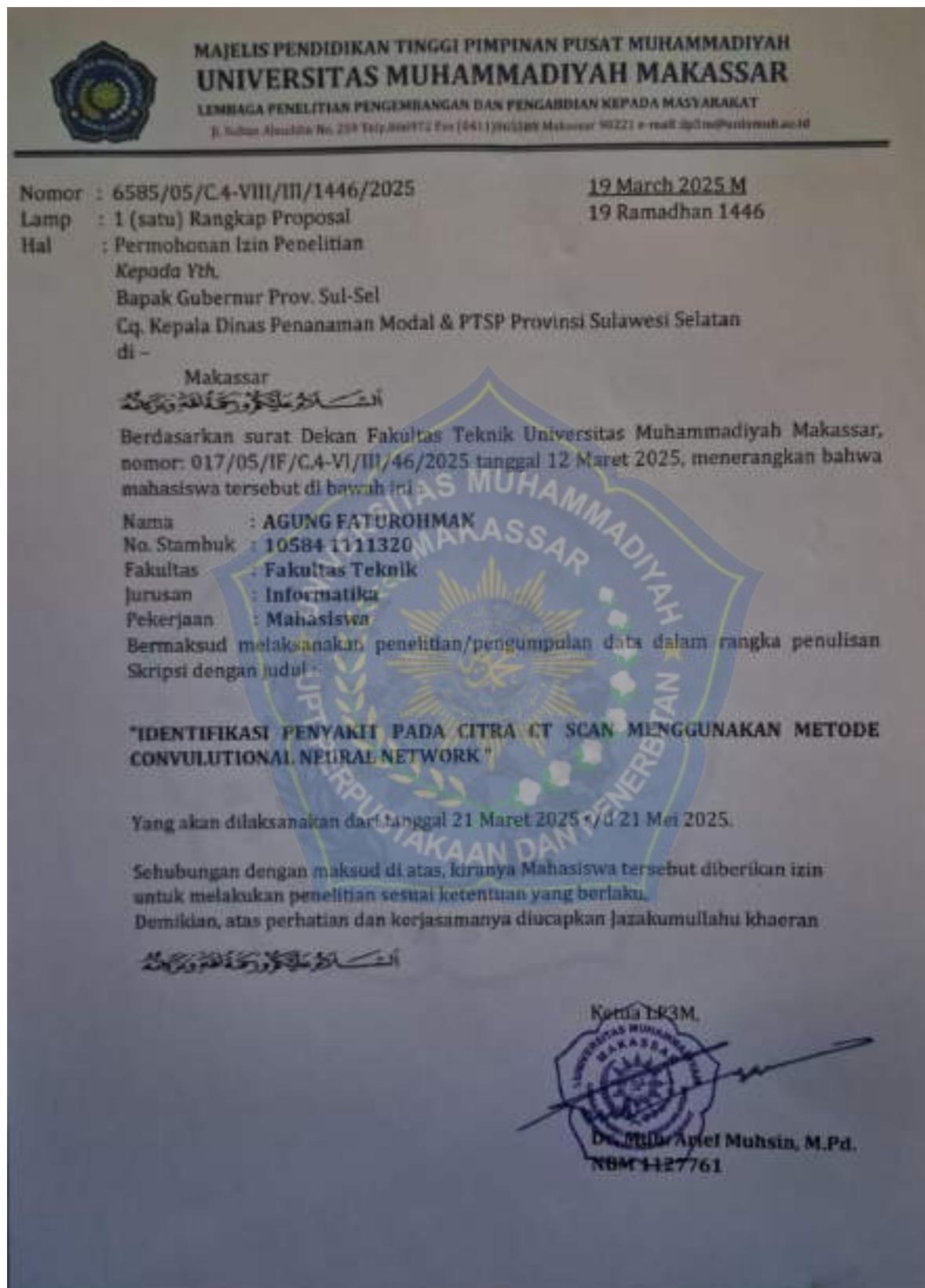
Tembusan:

1. Dekan Fakultas Teknik
2. Arsip

Gedung Mekarsari Lantai 3  
Jl. Sultan Abdurrahman No. 289 Telp. (0411) 866 972 Fax (0411) 866 988 Makassar 90231  
www.unismuh.ac.id [www.unismuh.ac.id](http://www.unismuh.ac.id) [lp3m.unismuh.ac.id](http://lp3m.unismuh.ac.id)

Kampus  
Merdeka

Lampiran 5. Surat Permohonan Izin Penelitian Kepada Kepala Dinas Penanaman Modal & PTSP Provinsi Sulawesi Selatan



Lampiran 6. Surat Izin Penelitian Dari PTSP Kepada RSUD Labuang Baji



PEMERINTAH PROVINSI SULAWESI SELATAN  
DINAS PENANAMAN MODAL DAN PELAYANAN TERPADU SATU PINTU

Jl. Bougainville No.5 Telp. (0411) 441077 Fax. (0411) 448836  
Website : <http://simap-new.sulselprov.go.id> Email : [ptsp@sulselprov.go.id](mailto:ptsp@sulselprov.go.id)  
Makassar 90231

Nomor : 8047/S.01/PTSP/2025

Kepada Yth.

Lampiran : -

Direktur RSUD Labuang Baji

Perihal : Izin penelitian

Makassar

di-  
Tempat

Berdasarkan surat Ketua LP3M UNISMUH Makassar Nomor. 6585/05/C.4-VIII/IV/1446/2025 tanggal 19 Maret 2025 perihal tersebut diatas, mahasiswa/peneliti dibawah ini:

Nama : AGUNG FATUROHMAN  
Nomor Pokok : 105841111320  
Program Studi : Teknik Informatika  
Pekerjaan/Lembaga : Mahasiswa (S1)  
Alamat : Jl. Siti Alauddin No. 259, Makassar



PROVINSI SULAWESI SELATAN

Bermaksud untuk melakukan penelitian di daerah/kantor saudara dalam rangka menyusun SKRIPSI, dengan judul :

**" IDENTIFIKASI PENYAKIT PADA CITRA CT SCAN MENGGUNAKAN METODE CONVOIUTIONAL NEURAI NETWORK "**

Yang akan dilaksanakan dari : Tgl. 24 April s/d 24 Juni 2025

Sehubungan dengan hal tersebut diatas, pada prinsipnya kami *menyetujui* kegiatan dimaksud dengan ketentuan yang tertera di belakang surat izin penelitian.

Demikian Surat Keterangan ini diberikan agar dipergunakan sebagaimana mestinya.

Diterbitkan di Makassar  
Pada Tanggal 24 April 2025

KEPALA DINAS PENANAMAN MODAL DAN PELAYANAN TERPADU  
SATU PINTU PROVINSI SULAWESI SELATAN



ASRUL SANI, S.H., M.Si.  
Pangkat : PEMBINA TINGKAT I  
Nip : 19750321 200312 1 008

Tembusan Yth:

1. Ketua LP3M UNISMUH Makassar di Makassar;
2. Pertiga

Lampiran 7. Surat Izin Penelitian RSUD Labuang Baji





**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 : Angung Faturohman**

**Nim : 10584111320**

**Program Studi : Teknik Informatika**

Dengan nilai:

No	Bab	Nilai	Ambang Batas
1	Bab 1	9%	10 %
2	Bab 2	24%	25 %
3	Bab 3	9%	10 %
4	Bab 4	4%	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, 22 Agustus 2025

Mengetahui,

Kepala UPT- Perpustakaan dan Penerbitan,

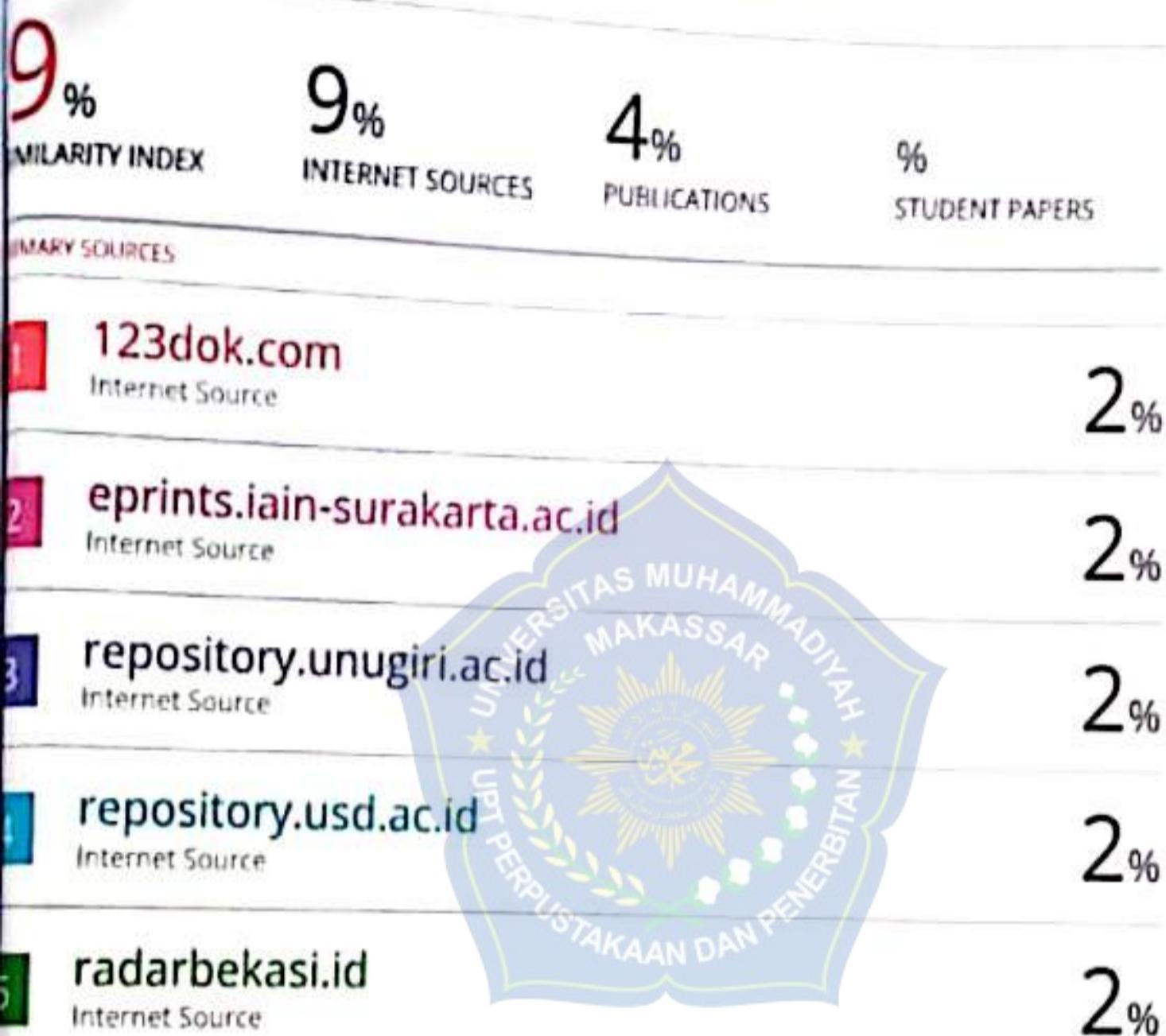


BAB I Agung faturohman  
105841111320

*by Tahap Tutup*



exion date: 22-Aug-2025 07:04AM (UTC+0700)  
ession ID: 2733123411  
ame: BAB\_1\_87.docx (20.84K)  
Count: 949  
User count: 6101



Include quotes

On

Exclude matches

< 2%

Include bibliography

On

# BAB II Agung faturohman

## 105841111320

by Tahap Tutup



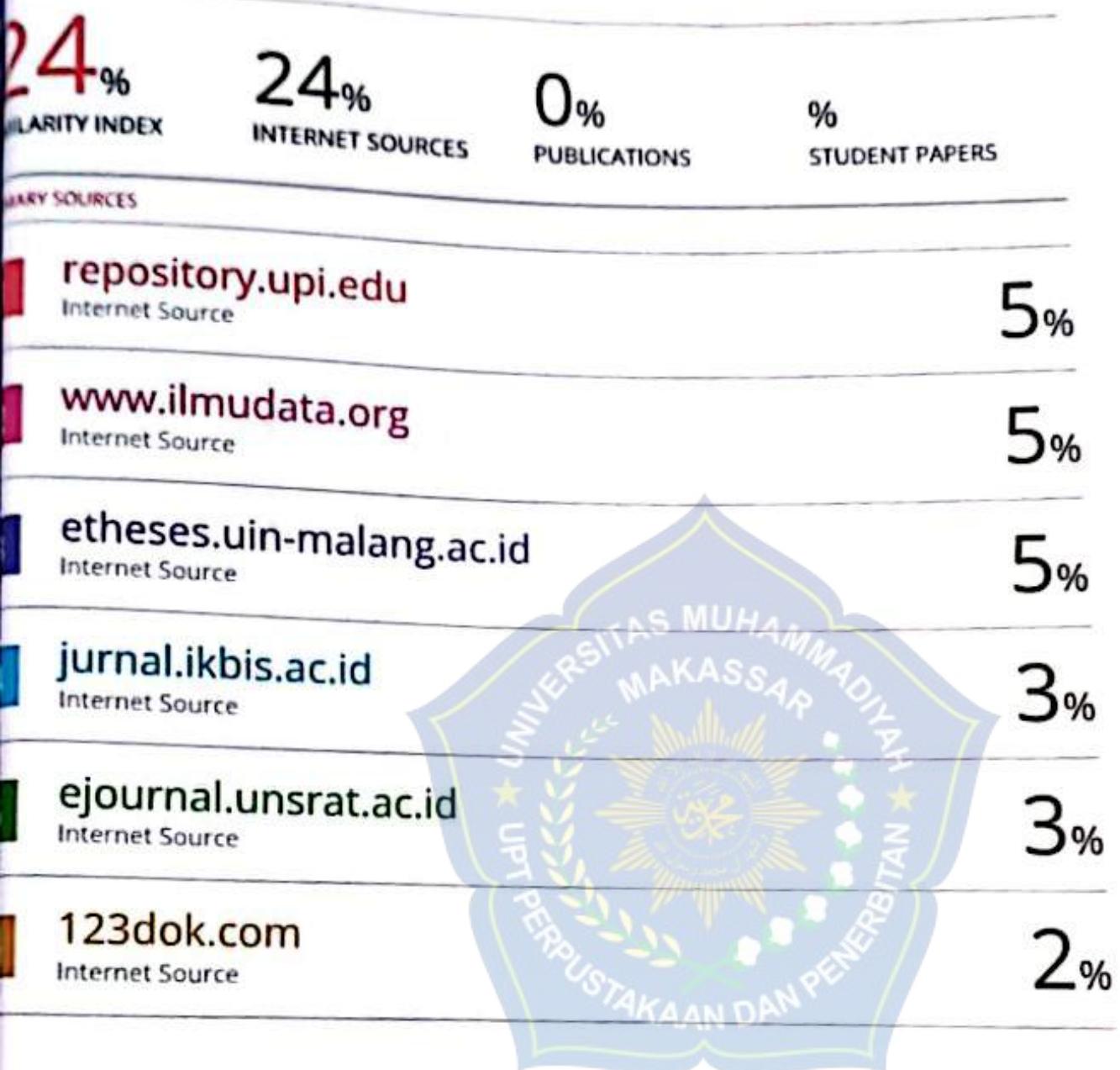
ession date: 21-Aug-2025 03:53PM (UTC+0700)

ession ID: 2732794261

ime: BAB\_II\_TINJAUAN\_PUSTAKA\_2.docx (312.66K)

count: 2512

acter count: 16217



Include quotes      On      Exclude matches      < 24%  
 Include bibliography      On

# BAB III Agung faturohman

## 105841111320

by Tahap Tutup



Creation date: 21-Aug-2025 03:54PM (UTC+0700)

Creation ID: 2732794540

File: BAB\_III\_AGUNGGG.docx (58.26K)

Page count: 1746

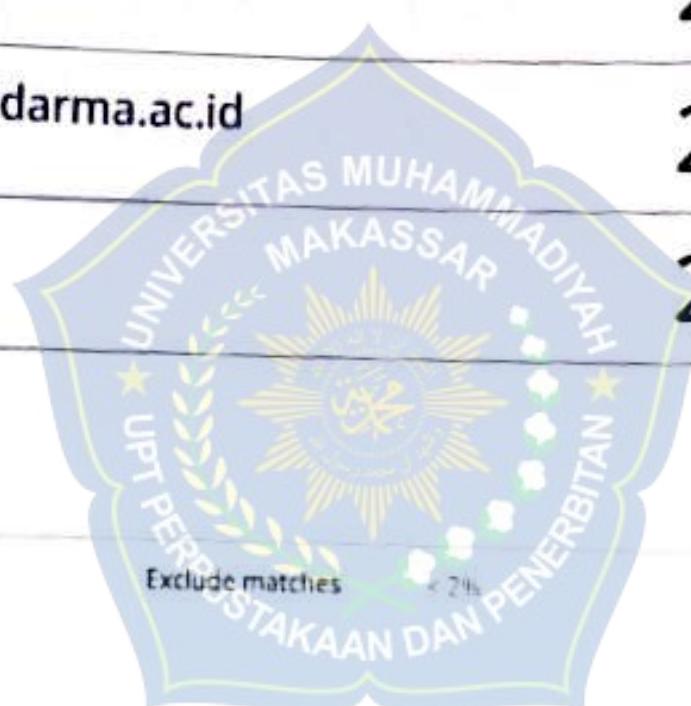
Character count: 11112

9%	MILARITY INDEX
9%	INTERNET SOURCES
3%	PUBLICATIONS

MARY SOURCES

 <a href="http://digilibadmin.unismuh.ac.id">digilibadmin.unismuh.ac.id</a> Internet Source	4%
 <a href="http://ejournal.sisfokomtek.org">ejournal.sisfokomtek.org</a> Internet Source	2%
 <a href="http://ejurnal.stmik-budidarma.ac.id">ejurnal.stmik-budidarma.ac.id</a> Internet Source	2%
 <a href="http://123dok.com">123dok.com</a> Internet Source	2%

clude quotes On  
clude bibliography On



Agung faturohman  
105841111320 BAB IV

*by Tahap Tutup*



ession date: 19-Aug-2025 11:26AM (UTC+0700)

ession ID: 2731742593

me: BAB\_IV\_3.docx (1.9M)

ount: 2122

ter count: 13467



- 1 Doly Ilham Saputra Huta Julu, Dewi Nurdiyah. "KLASIFIKASI SAMPAH ORGANIK DAN NON ORGANIK MENGGUNAKAN TRANSFER LEARNING", Jurnal Transformatika, 2025  
Publication 1 %
- 2 inass.org 1 %  
Internet Source
- 3 Submitted to UM Surabaya 1 %  
Student Paper
- 4 Akram Kemal Dewantara. "Optimasi Pengambilan Keputusan dengan Neural Network: Menuju Era Keputusan Pintar", The Indonesian Journal of Computer Science, 2024  
Publication 1 %
- 5 id.123dok.com 1 %  
Internet Source
- 6 Submitted to Universitas Esa Unggul 1 %  
Student Paper
- 7 etheses.uin-malang.ac.id 1 %  
Internet Source
- 8 Shona Chayy Bilqisth, Rohmatulloh Muhamad Ikhsanuddin. "ANALISIS PERBANDINGAN AKURASI KLASIFIKASI KEPUASAN SISWA TERHADAP KINERJA GURU MENGGUNAKAN ALGORITMA SVM, C4.5 DAN RANDOM 1 %

# FOREST", E-Link: Jurnal Teknik Elektro dan Informatika, 2025

Publication

Include quotes On      Exclude matches < 1%

Include bibliography On



Agung faturohman  
105841111320 BAB V



Version date: 19-Aug-2025 11:27AM (UTC+0700)

Version ID: 2731742879

Name: BAB\_V\_5.docx (14.55K)

Page count: 185

Character count: 1222

5%  
SIMILARITY INDEX

5%  
INTERNET SOURCES

0%  
PUBLICATIONS

0%  
STUDENT PAPERS

PRIMARY SOURCES

docobook.com  
Internet Source

5%

Include quotes Off  
Include bibliography Off



